MSC GEODETIC ENGINEERING

MSR-02: ADVANCED TECHNIQUES FOR MOBILE SENSING AND ROBOTICS (GEODESY TRACK)

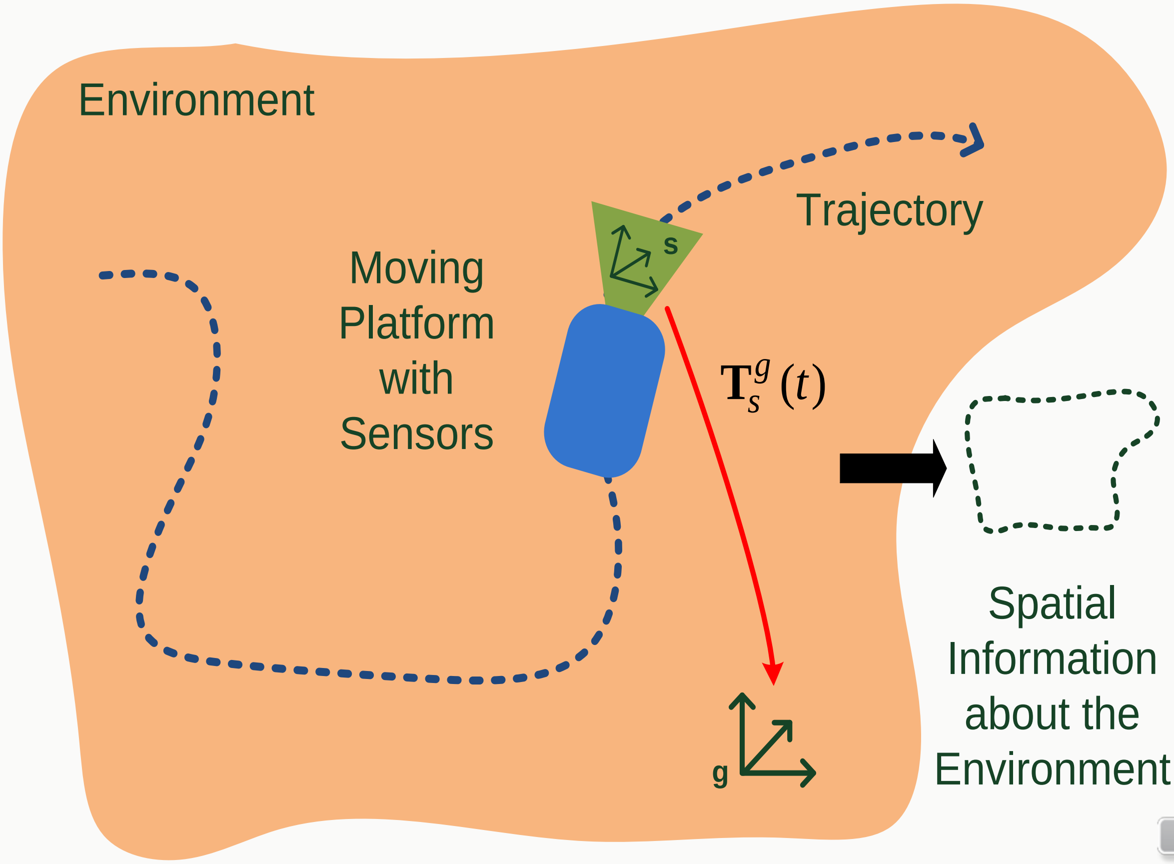
02: TRAJECTORY ESTIMATION

DR. LASSE KLINGBEIL

INSTITUTE FOR GEODESY AND GEOINFORMATION
UNIVERSITY OF BONN

ADVANCED TECHNIQUES FOR MOBILE SENSING AND ROBOTICS – LECTURE CONTENT

MOBILE MAPPING



CHAPTER 1: MOBILE LASER SCANNING

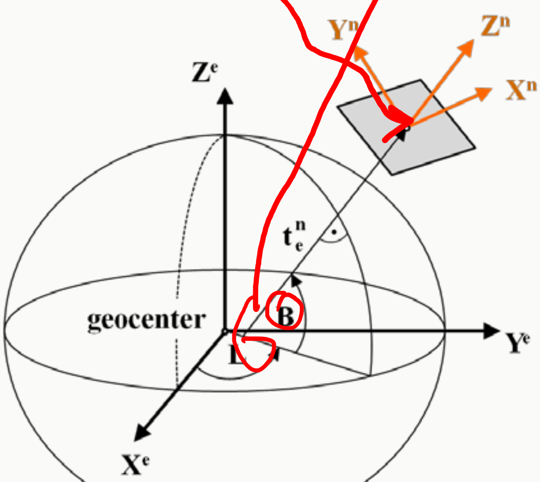
$$\mathbf{p}^{global}_{object}(t_s) \quad = \quad \mathbf{T}^{global}_{body}(t_s) \cdot \mathbf{T}^{body}_{sensor} \cdot \mathbf{p}^{sensor}_{object}(t_s)$$

- Review of involved **coordinate systems /frames**
- Derivation of detailed **georeferencing equation** for the example of **mobile laser scanning**

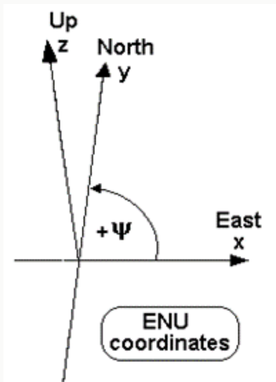
$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \mathbf{R}^e_n(L,B)\,\mathbf{R}^n_b(\phi,\theta,\psi) \cdot \left[\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + \mathbf{R}^b_s(\alpha,\beta,\gamma) \cdot \begin{bmatrix} 0 \\ d \cdot \sin b \\ d \cdot \cos b \end{bmatrix} \right]$$

CHAPTER 2: TRAJECTORY PARAMETERS

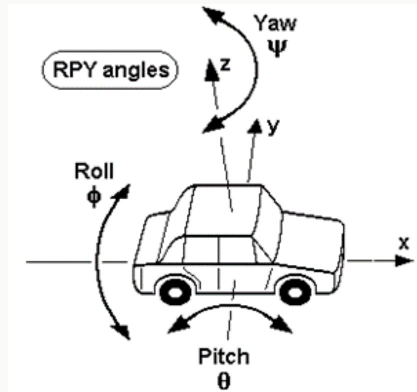
$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \mathbf{R}_n^e\,(L, B)\,\mathbf{R}_b^n\,(\phi, \theta, \psi) \cdot \left[\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + \mathbf{R}_s^b\,(\alpha, \beta, \gamma) \cdot \begin{bmatrix} 0 \\ d \cdot \sin b \\ d \cdot \cos b \end{bmatrix} \right]$$



Earth frame

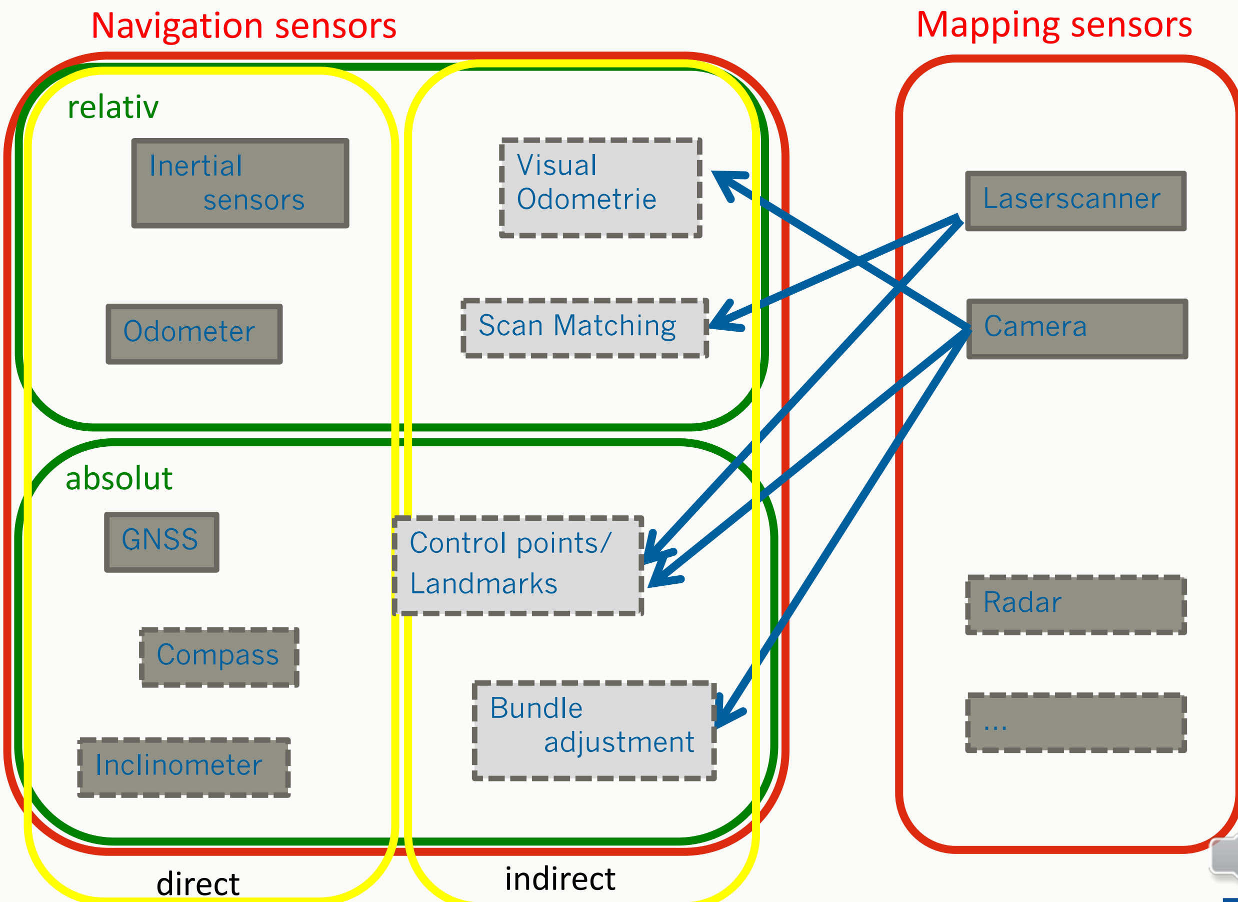


Navigation frame



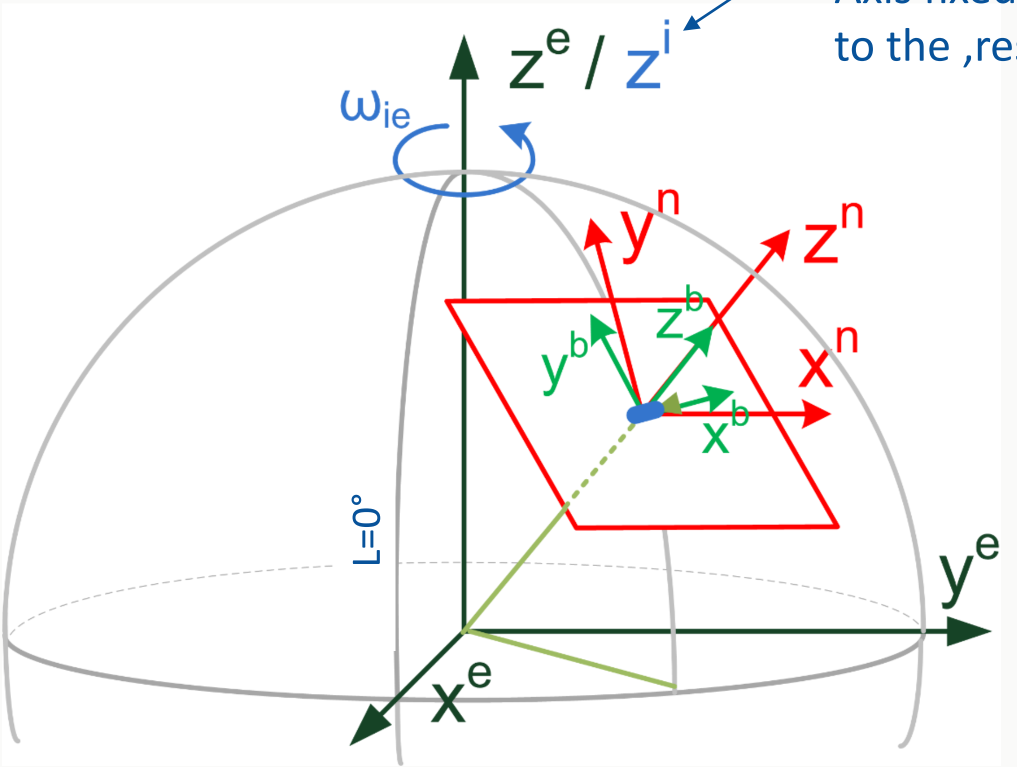
"body frame"

SENSORS OVERVIEW



INERTIAL SENSORS

COORDINATE SYSTEMS



Inertial frame

Axis fixed with respect to the ‚rest of the universe'

UNIVERSITÄT BONN igg

INERTIAL SENSORS

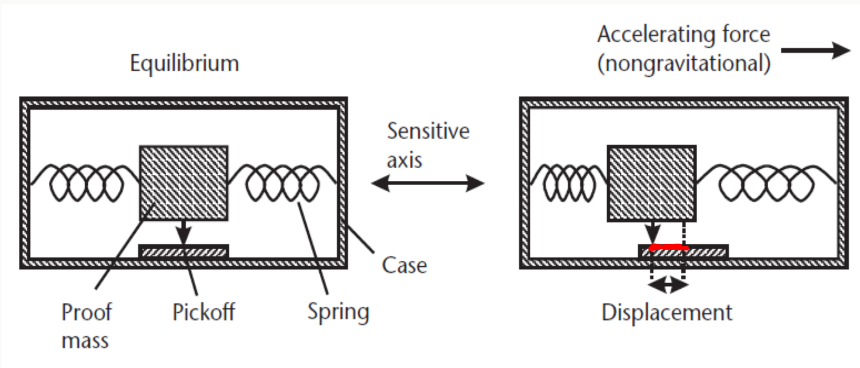
ACCELEROMETER

- Measurement of specific force
  (= non gravitational acceleration)

$F = 0$

$a = 9{,}81\ m/s^2$

$g = -9{,}81\ m/s^2$



Specific force of the body frame relativ to the inertial frame, given in body frame coordinates

gravitational acceleration

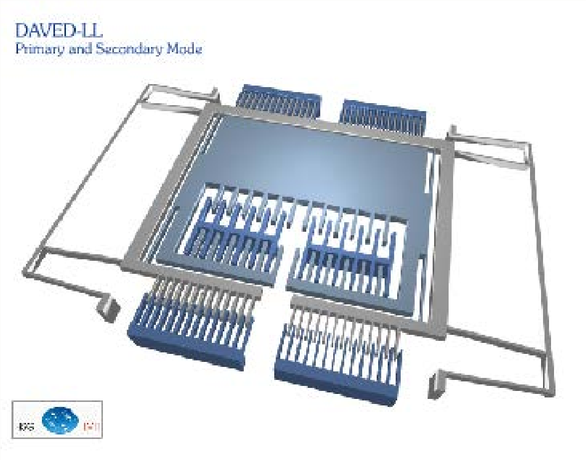
$$\mathbf{f}_{ib}^{b} = \underline{\mathbf{a}_{ib}^{b}} - \gamma_{ib}^{b}$$

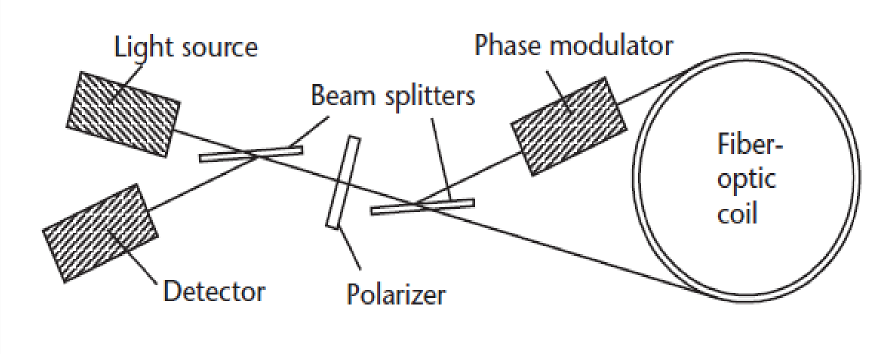
INERTIAL SENSORS

GYROSCOPE

- Measure angular rate of the body relative to the inertial frame given in body frame coordinates

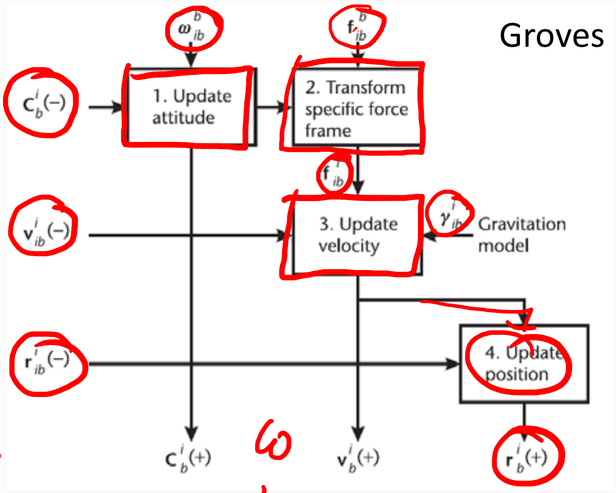
$$\omega_{ib}^{b}$$





STRAP DOWN INTEGRATION

INERTIAL FRAME EQUATIONS



- Use inertial frame as the system of resolving frame for position and orientation

- Update orientation

Integration errors

$$\mathbf{C}_b^i(+) \approx \mathbf{C}_b^i(-)(\mathbb{1} + \boldsymbol{\Omega}_{ib}^b \tau_i)$$

- Transform accelerometer measurement to inertial frame

$$\mathbf{f}_{ib}^i = \mathbf{C}_b^i \mathbf{f}_{ib}^b$$

- Correct for gravity

$$\mathbf{a}_{ib}^i = \mathbf{f}_{ib}^i + \gamma_{ib}^i$$

- Update velocity

$$\mathbf{v}_{ib}^i(+) \approx \mathbf{v}_{ib}^i(-) + \mathbf{a}_{ib}^i \tau_i$$

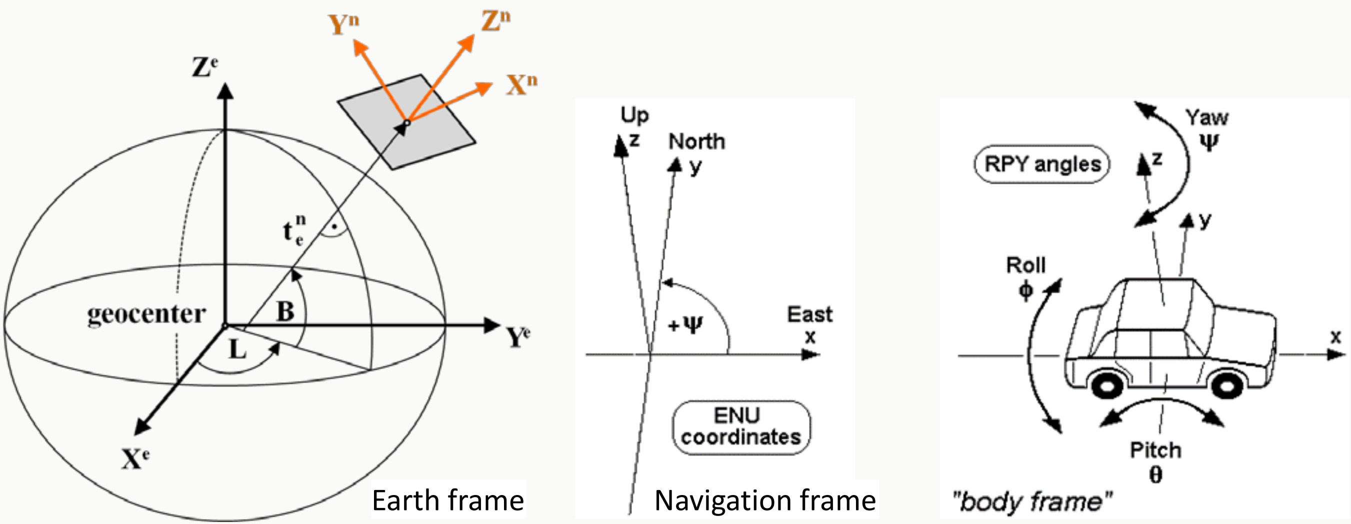
- Update position

$$\mathbf{r}_{ib}^i(+) \approx \mathbf{r}_{ib}^i(-) + \mathbf{v}_{ib}^i(-) \tau_i$$

UNIVERSITÄT BONN igg

TRAJECTORY PARAMETERS

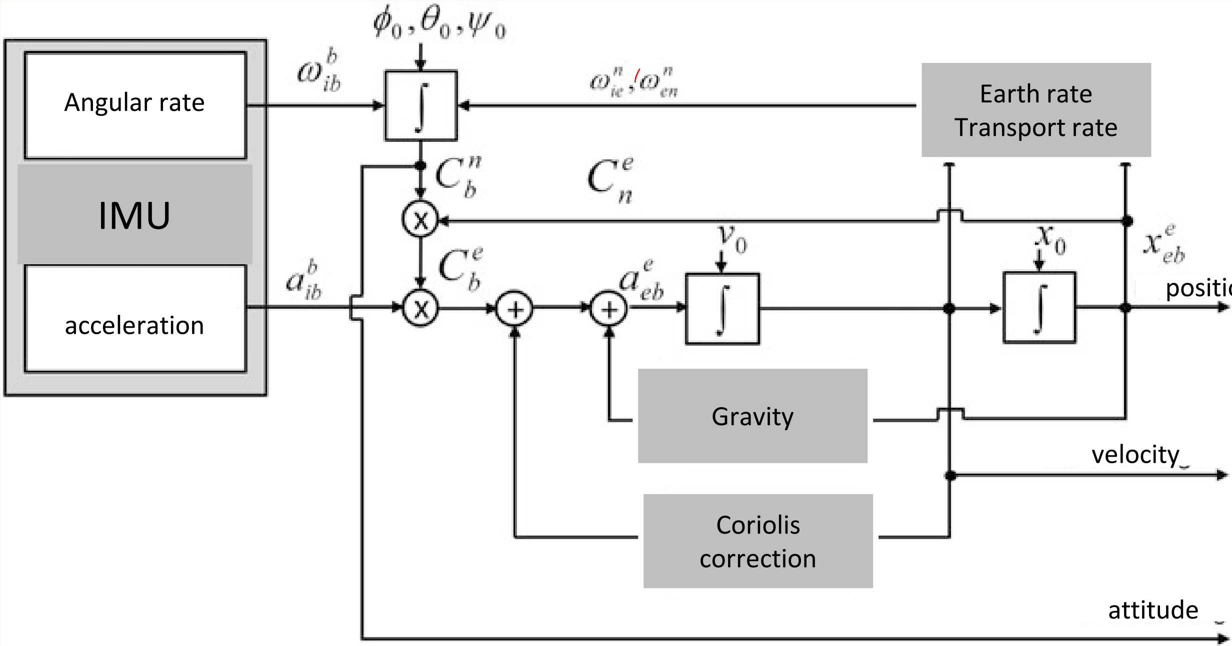
$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \mathbf{R}_n^e(L,B)\,\mathbf{R}_b^n(\phi,\theta,\psi)\cdot\left[\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + \mathbf{R}_s^b(\alpha,\beta,\gamma)\cdot\begin{bmatrix} 0 \\ d\cdot\sin b \\ d\cdot\cos b \end{bmatrix}\right]$$



STRAPDOWN INTEGRATION

WANTED:

$$\begin{pmatrix} \mathbf{q}_b^n \\ \mathbf{p}_{eb}^e \\ \mathbf{v}_{eb}^n \end{pmatrix}_k = f\left(\begin{pmatrix} \mathbf{q}_b^n \\ \mathbf{p}_{eb}^e \\ \mathbf{v}_{eb}^n \end{pmatrix}_{k-1}, \begin{pmatrix} \mathbf{a}_{ib}^b \\ \boldsymbol{\omega}_{ib}^b \end{pmatrix}_{k-1} \right)$$



UNIVERSITÄT BONN igg

INITIALIZATION AND ALIGNEMENT

HOW TO GET THE STARTING VALUES?

- Position
  - Use external systems, such as GNSS
  - Start from a known position (e.g. car has parked at known position)
- Orientation
  - Accelerometer levelling
  - Gyro Compassing
  - Magnetometer/Compass
  - Multiple GNSS Antennas
- Velocity
  - Use external system (e.g. GNSS)
  - Start in a static situation (v=0)

INS ERROR PROPAGATION

Orientation update $$\mathbf{C}_b^i(+) \approx \mathbf{C}_b^i(-)(\mathbb{1} + \mathbf{\Omega}_{ib}^b \tau_i)$$

- Constant gyroscope error leads to orientation error

$$\delta\phi_b^n(t) = \mathbf{b}_\omega t$$

  - e.g. 0.1 ° gyro bias → 30° orientation error after 5 minutes

UNIVERSITÄT BONN igg

INS ERROR PROPAGATION

Position/velocity update

$$\mathbf{v}_{ib}^{i}(+) \approx \mathbf{v}_{ib}^{i}(-) + \mathbf{a}_{ib}^{i}\tau_i$$
$$\mathbf{r}_{ib}^{i}(+) \approx \mathbf{r}_{ib}^{i}(-) + \mathbf{v}_{ib}^{i}(-)\tau_i$$

- Constant velocity error integrates to position error

$$\delta\mathbf{r}_{eb}^{n}(t) \approx \delta\mathbf{v}_{eb}^{n}t$$

  - e.g. 0.1 m/s velocity error ➔ 30m position error after 5 minutes

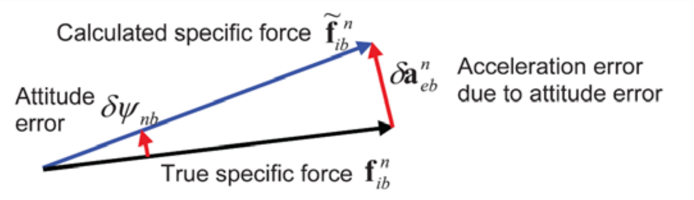
- Constant accelerometer bias leads to velocity and position errors

$$\delta\mathbf{v}_{eb}^{n}(t) \approx \mathbf{C}_{b}^{n}\mathbf{b}_a t, \qquad \delta\mathbf{r}_{eb}^{n}(t) \approx \frac{1}{2}\mathbf{C}_{b}^{n}\mathbf{b}_a t^2$$

  - e.g. 0.01 m/s^2 accelerometer bias ➔ 450m position error after 5 minutes

INS ERROR PROPAGATION

- Constant attitude errors lead to wrong transformations of the measured specific-force into the resoving frame
  - „Gravity compensation“ is wrong, parts of the measured reaction to gravity are considered as acceleration leading to acceleration offsets



  - e.g. 0.057° attitude error ➔ 440m position error after 5 minutes
- Constant gyroscope bias leads to attitude error leads to position error

$$\delta\mathbf{r}^n_{eb}(t) \propto t^3$$

  - e.g. 2.1°/hr bias ➔ 440m position error after 5 minutes

UNIVERSITÄT BONN igg

INS ERROR PROPAGATION

CONSTANT ERRORS

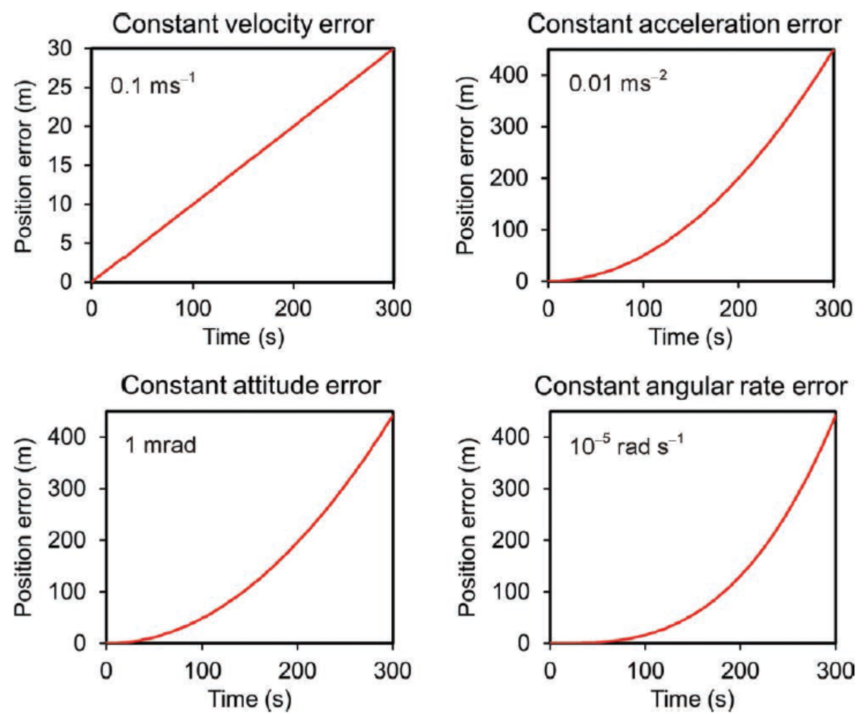


Fig. 10.
Short-term straight-line position error growth per axis for different error sources. (From [3] © Paul Groves 2013. Reproduced with permission.)

UNIVERSITÄT BONN igg

INS ERROR PROPAGATION

SENSOR NOISE

- Noisy measurements of accelerometers and gyroscopes also lead to noisy position errors (uncertainty)

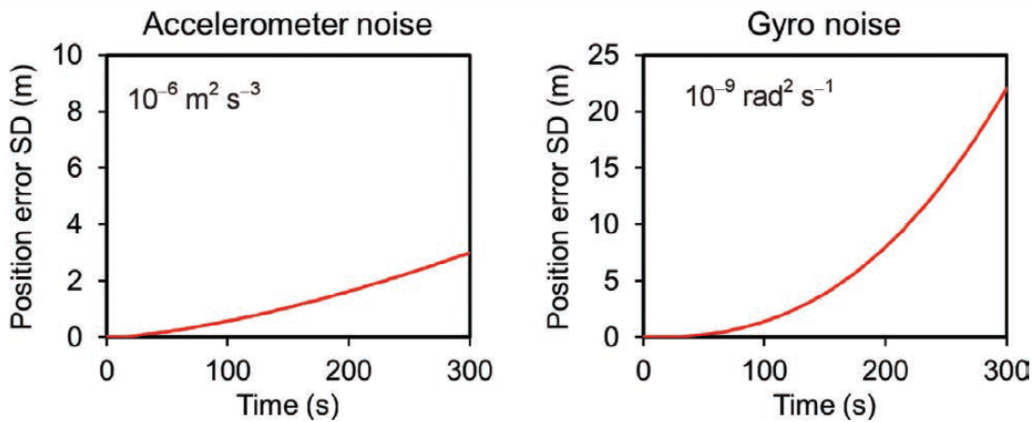


Fig. 12.
Short-term straight-line position error standard deviation growth per axis because of inertial sensor noise. (From [3] © Paul Groves 2013. Reproduced with permission.)

UNIVERSITÄT BONN igg

NON-LINEAR FILTERING

- Linear Kalman Filter:
  - linear system and measurement models

$$\mathbf{x}_k = \mathbf{\Phi}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_k + \mathbf{G}\mathbf{w}_{s,k}$$
$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{w}_{m,k}$$

- General case:
  - predicted state is a function of last state, control input and noise

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_{s,k})$$

  - Measurement is a function of the state

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{w}_{m,k})$$

KALMAN FILTER

Predicted estimate

Predicted covariance

System noise

$$\mathbf{x}_k^- = \boldsymbol{\Phi}\mathbf{x}_{k-1}$$

$$P_k^- = \boldsymbol{\Phi} P_{k-1}\boldsymbol{\Phi}^T + \mathbf{Q}$$

Prediction

measurement noise

Kalman Gain

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

$$\mathbf{x}_k = \mathbf{x}_k^- + K_k(z_k - H\mathbf{x}_k^-)$$

$$P_k = (I - K_k H)P_k^-$$

Correction

measurement

Predicted measurement

Updated estimate

Updated covariance

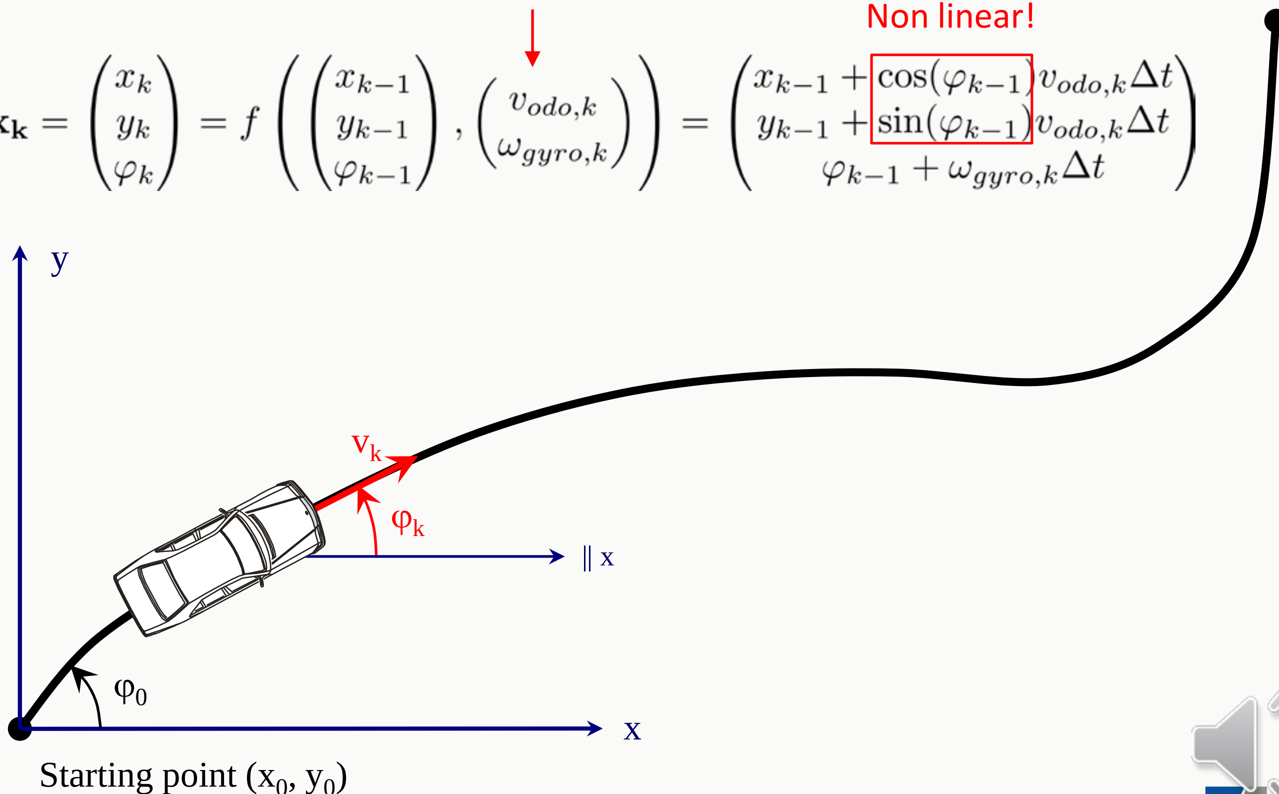
NON LINEAR EXAMPLE (DEAD RECKONING)

SYSTEM MODEL

control input

Non linear!

$$\mathbf{x_k} = \begin{pmatrix} x_k \\ y_k \\ \varphi_k \end{pmatrix} = f\left(\begin{pmatrix} x_{k-1} \\ y_{k-1} \\ \varphi_{k-1} \end{pmatrix}, \begin{pmatrix} v_{odo,k} \\ \omega_{gyro,k} \end{pmatrix} \right) = \begin{pmatrix} x_{k-1} + \boxed{\cos(\varphi_{k-1})} v_{odo,k} \Delta t \\ y_{k-1} + \boxed{\sin(\varphi_{k-1})} v_{odo,k} \Delta t \\ \varphi_{k-1} + \omega_{gyro,k} \Delta t \end{pmatrix}$$

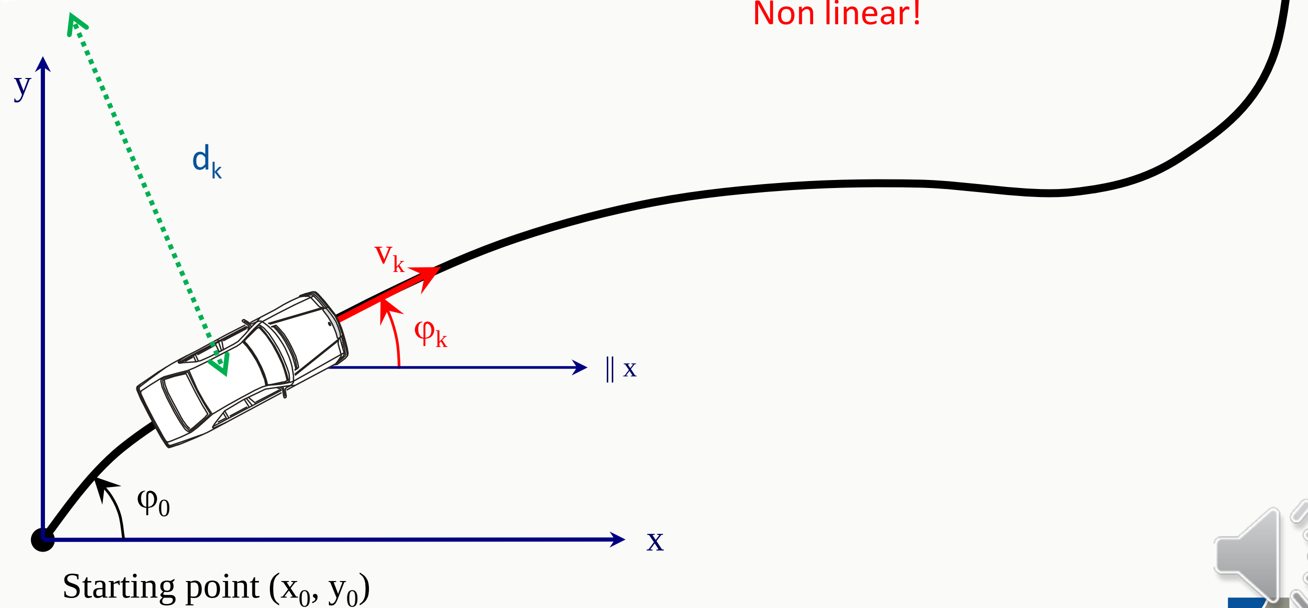


NON LINEAR EXAMPLE (DEAD RECKONING)

MEASUREMENT MODEL (E.G. GPS, DISTANCE TO KNOWN POSITION)

$$z_k = d_k = h(\mathbf{x}_k) = \boxed{\sqrt{(x_k - x_{sat,k})^2 + (y_k - y_{sat,k})^2}}$$

Non linear!



SUMMARY EXTENDED KALMAN FILTER

Prediction

$$
\begin{aligned}
\mathbf{x}_k^- &= f(\mathbf{x}_{k-1}, \mathbf{u}_k) \\
\mathbf{P}_k^- &= \mathbf{\Phi} \mathbf{P}_{k-1} \mathbf{\Phi}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T
\end{aligned}
$$

Correction

$$
\begin{aligned}
\mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \\
\mathbf{x}_k &= \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{z}_k - h(\mathbf{x}_k^-)) \\
\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H}) \mathbf{P}_k^-
\end{aligned}
$$

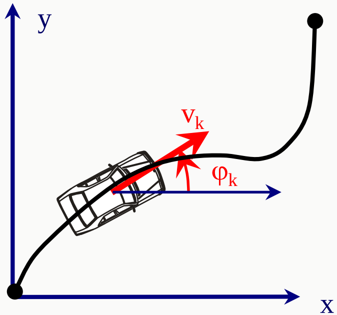
$$
\mathbf{\Phi} = \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}, \mathbf{u}=\mathbf{u}_k}, \quad \mathbf{G} = \frac{\partial f(\mathbf{x}, \mathbf{u})}{\partial \mathbf{w}} \bigg|_{\dots}
$$

$$
\mathbf{H}_k = \frac{\partial h(\mathbf{x})}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\hat{\mathbf{x}}_k^-}
$$

2D KF EXAMPLE

SYSTEM MODEL

- No assumed (and wrong) model on the temporal evolutionof motion states
- Relative measurements (path increment and angle change) are modelled as control input
- System noise is represented as control noise



control input

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \boxed{\mathbf{u}_k}, \boxed{\mathbf{w}_k})$$

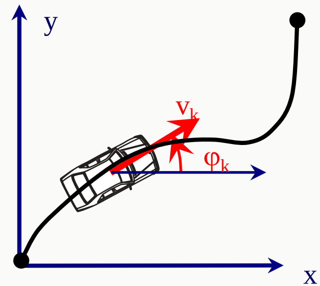
$$\begin{bmatrix} x_k \\ y_k \\ \varphi_k \end{bmatrix} = f\left(\begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \varphi_{k-1} \end{bmatrix}, \begin{bmatrix} \Delta s_k \\ \omega_k \end{bmatrix}, \begin{bmatrix} w_{odo} \\ w_{gyro} \end{bmatrix} \right) = \begin{bmatrix} x_{k-1} + \cos(\varphi_{k-1})(\boxed{\Delta s_k} \boxed{+ w_{odo}}) \\ y_{k-1} + \sin(\varphi_{k-1})(\boxed{\Delta s_k} \boxed{+ w_{odo}}) \\ \varphi_{k-1} + (\boxed{\omega_{gyro,k}} + \boxed{w_{gyro}})\Delta t \end{bmatrix}$$

System/control noise

2D KF EXAMPLE

TRANSITION MATRIX

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \varphi \end{bmatrix}$$



$$\boldsymbol{\Phi}_{k-1} = \frac{\partial}{\partial \mathbf{x}} \begin{bmatrix} x_{k-1} + \cos(\varphi_{k-1})(\Delta s_k + w_{odo}) \\ y_{k-1} + \sin(\varphi_{k-1})(\Delta s_k + w_{odo}) \\ \varphi_{k-1} + (\omega_{gyro,k} + w_{gyro})\Delta t \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}_{k-1},\mathbf{u}=\mathbf{u}_k}$$

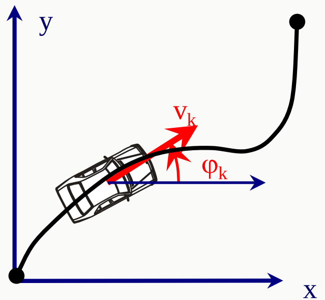
$$= \begin{bmatrix} 1 & 0 & -\sin(\varphi_{k-1})\Delta s_k \\ 0 & 1 & \cos(\varphi_{k-1})\Delta s_k \\ 0 & 0 & 1 \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}_{k-1},\mathbf{u}=\mathbf{u}_k}$$

2D KF EXAMPLE

$$\begin{aligned} \mathbf{x}_k^- &= f(\mathbf{x}_{k-1}, \mathbf{u}_k) \\ \mathbf{P}_k^- &= \mathbf{\Phi} \mathbf{P}_{k-1} \mathbf{\Phi}^T + \boxed{\mathbf{G}\mathbf{Q}\mathbf{G}^T} \end{aligned}$$

SYSTEM NOISE

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \varphi \end{bmatrix}$$



$$\mathbf{Q} = \begin{bmatrix} \sigma_{odo}^2 & 0 \\ 0 & \sigma_{gyro}^2 \end{bmatrix}$$

$$\mathbf{G} = \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{w})}{\partial \mathbf{w}} = \begin{bmatrix} \cos(\varphi_{k-1}) & 0 \\ \sin(\varphi_{k-1}) & 0 \\ 0 & \Delta t \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}, \mathbf{u}=\mathbf{u}_k}$$

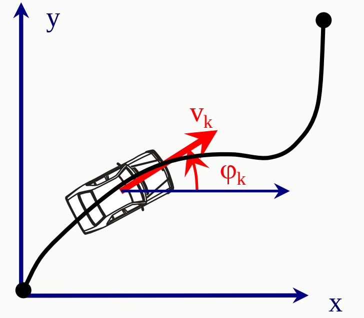
$$\mathbf{G}\mathbf{Q}\mathbf{G}^T = \begin{bmatrix} \cos^2(\varphi_{k-1})\sigma_{odo}^2 & \cos(\varphi_{k-1})\sin(\varphi_{k-1})\sigma_{odo}^2 & 0 \\ \cos(\varphi_{k-1})\sin(\varphi_{k-1})\sigma_{odo}^2 & \sin^2(\varphi_{k-1})\sigma_{odo}^2 & 0 \\ 0 & 0 & \sigma_{gyro}^2 \Delta t^2 \end{bmatrix}$$

UNIVERSITÄT BONN igg

2D KF EXAMPLE

MEASUREMENT MODEL

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \varphi \end{bmatrix}$$



$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \boxed{\mathbf{R}})^{-1} \\ \mathbf{x}_k &= \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{z}_k - \boxed{h(\mathbf{x}_k^-)}) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \end{aligned}$$

- Measurement is only GPS Position x_{gps}, y_{gps}

$$\mathbf{z}_k = \begin{bmatrix} x_{gps,k} \\ y_{gps,k} \end{bmatrix} = h(\mathbf{x}_k) = \mathbf{H}\mathbf{x}_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{x}_k$$

- Measurement Noise

$$\mathbf{R} = \begin{bmatrix} \sigma^2_{gps,x} & 0 \\ 0 & \sigma^2_{gps,y} \end{bmatrix}$$

GPS/IMU INTEGRATION

- Estimate the position and orientation of the system in all dimensions
- Use inertial sensors (acc + gyro) as control input: Strapdown inertial integration as system model
- Because acc + gyro have time dependant systematic errors (bias), estimate the biases as part of the state vector
- Use GPS as measurements
  - Use of positions: ➔ loosely coupled integration
  - Use of GPS raw data (double differences, carrier phases) ➔ tightly-coupled integration

GPS/IMU INTEGRATION

- State Vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_b^n \\ \mathbf{x}_{eb}^e \\ \mathbf{v}_{eb}^n \\ \mathbf{b}_{gyro} \\ \mathbf{b}_{acc} \end{bmatrix}$$

Orientation quaternion

IMU bias values

- Control input

$$\mathbf{u} = \begin{bmatrix} \mathbf{f}_{ib}^b \\ \boldsymbol{\omega}_{ib}^b \end{bmatrix}$$

IMU sensor readings

- Measurement $\mathbf{z} = \mathbf{x}_{gps}$

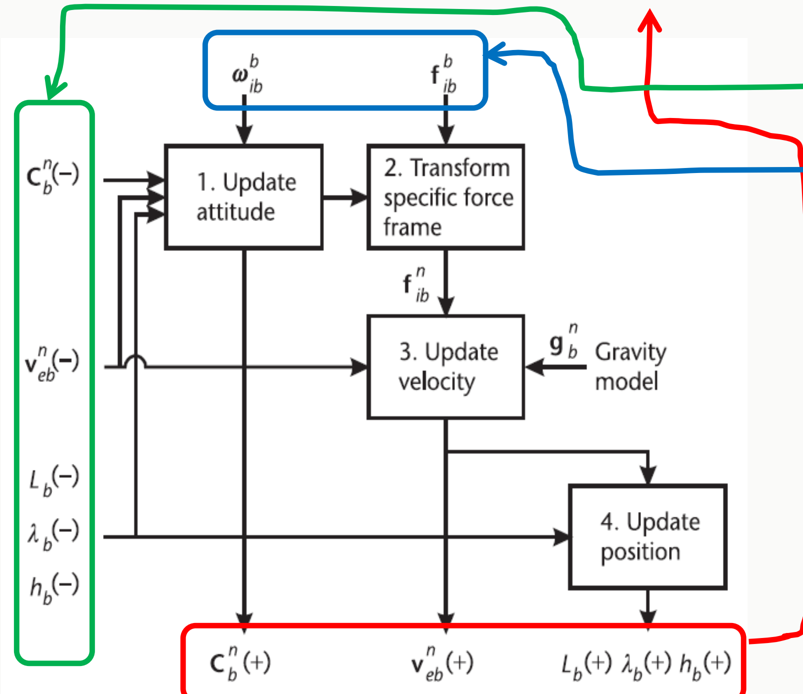
Processed GNSS position

GPS/IMU INTEGRATION

SYSTEM MODEL
STRAPDOWN ALGORITHM

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k, \mathbf{w}_k)$$

$$\begin{bmatrix} \mathbf{q}_b^n \\ \mathbf{p}_{eb}^e \\ \mathbf{v}_{eb}^n \\ \mathbf{b}_{gyro} \\ \mathbf{b}_{acc} \end{bmatrix}_k = f\left(\begin{bmatrix} \mathbf{q}_b^n \\ \mathbf{p}_{eb}^e \\ \mathbf{v}_{eb}^n \\ \mathbf{b}_{gyro} \\ \mathbf{b}_{acc} \end{bmatrix}_k, \begin{bmatrix} \mathbf{a}_{ib}^b \\ \boldsymbol{\omega}_{ib}^b \end{bmatrix}_k, \begin{bmatrix} \mathbf{w}_\omega \\ \mathbf{w}_a \end{bmatrix}_k \right)$$



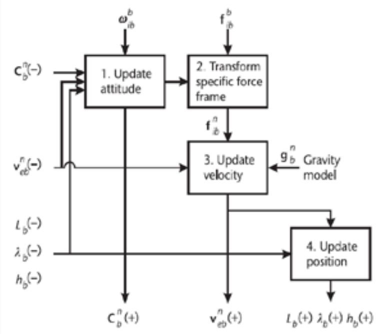
System noise is IMU noise

UNIVERSITÄT BONN igg

GPS/IMU INTEGRATION

MEASUREMENT MODEL

$$\mathbf{x} = \begin{bmatrix} \mathbf{q}_b^n \\ \mathbf{x}_{eb}^e \\ \mathbf{v}_{eb}^n \\ \mathbf{b}_{gyro} \\ \mathbf{b}_{acc} \end{bmatrix}$$



$$\begin{aligned} \mathbf{K}_k &= \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \boxed{\mathbf{R}})^{-1} \\ \mathbf{x}_k &= \mathbf{x}_k^- + \mathbf{K}_k(\mathbf{z}_k - \boxed{h(\mathbf{x}_k^-)}) \\ \mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k \mathbf{H})\mathbf{P}_k^- \end{aligned}$$

- Measurement is only GPS Position $x_{gps}, y_{gps}, z_{gps}$

$$\mathbf{z}_k = \begin{bmatrix} x_{gps,k} \\ y_{gps,k} \\ z_{gps,k} \end{bmatrix} = h(\mathbf{x}_k) = \mathbf{H}\mathbf{x}_k = \begin{bmatrix} \dots & 1 & 0 & 0 & \dots \\ \dots & 0 & 1 & 0 & \dots \\ \dots & 0 & 0 & 1 & \dots \end{bmatrix} \mathbf{x}_k$$

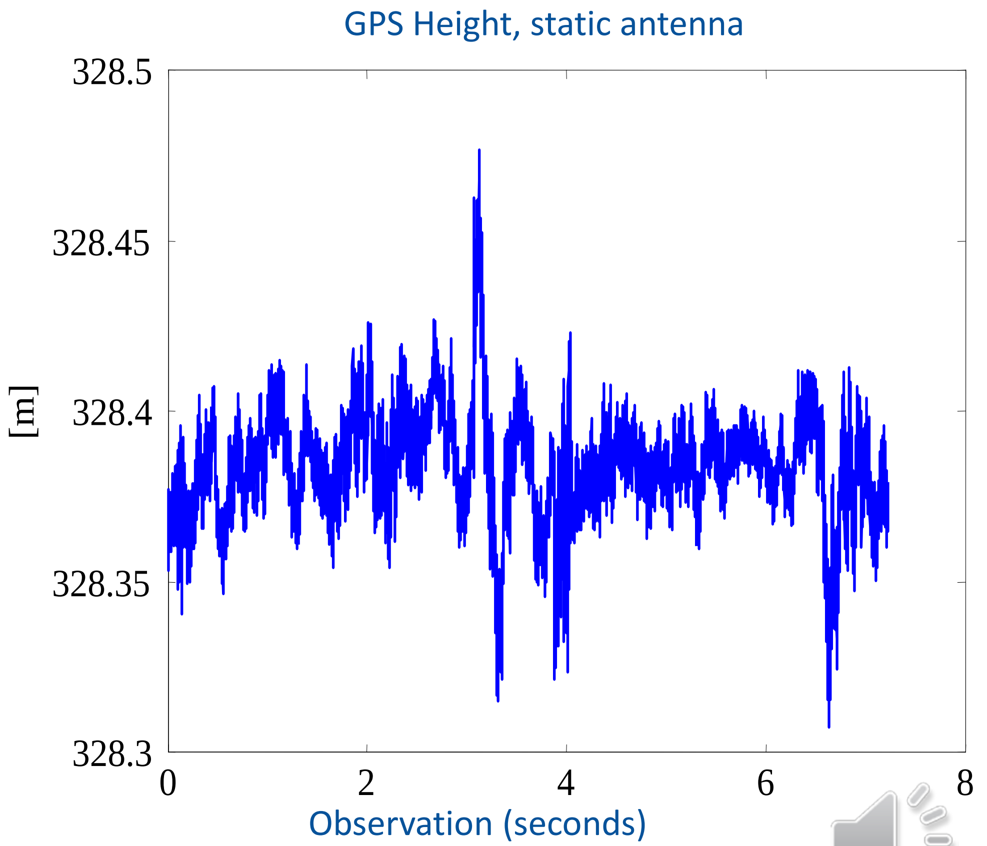
- Measurement Noise

$$\mathbf{R} = \begin{bmatrix} \sigma_{gps,x}^2 & 0 & 0 \\ 0 & \sigma_{gps,y}^2 & 0 \\ 0 & 0 & \sigma_{gps,z}^2 \end{bmatrix}$$

GPS/IMU INTEGRATION

GPS NOISE

Carrierphase
2-Frequency
Differential
Kinematic
Post-Processing

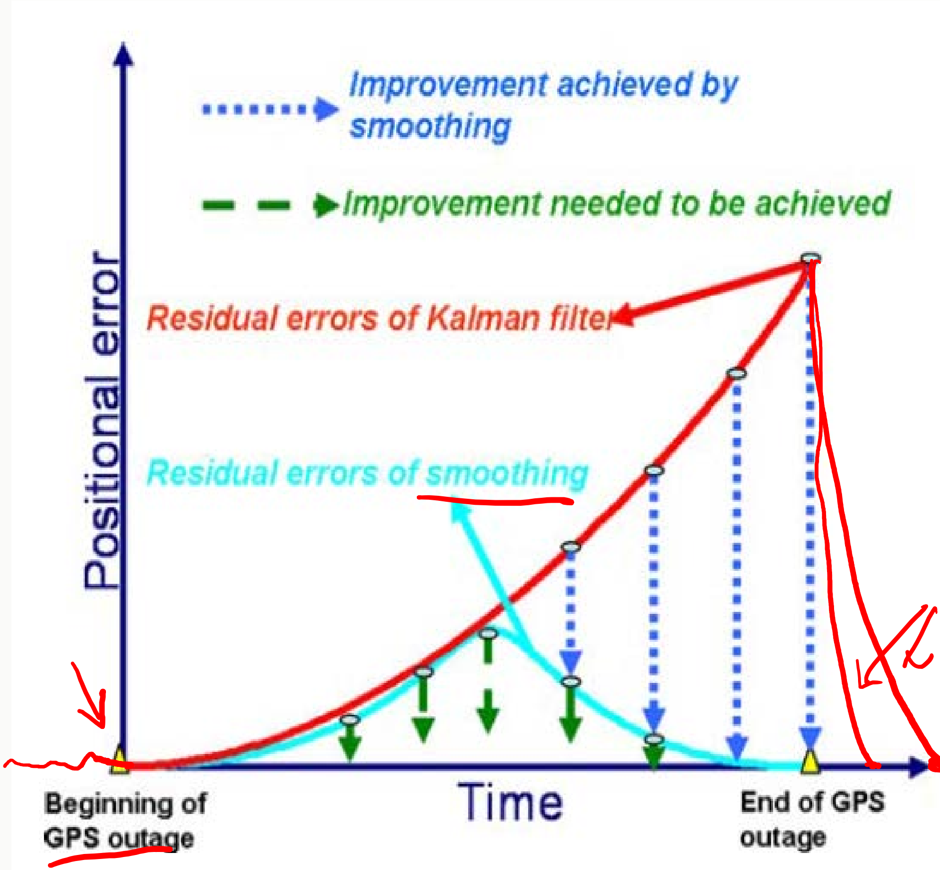


Remember

GPS noise is not white!

$$\mathbf{R} = \begin{bmatrix} \sigma^2_{gps,x} & 0 & 0 \\ 0 & \sigma^2_{gps,y} & 0 \\ 0 & 0 & \sigma^2_{gps,z} \end{bmatrix}$$

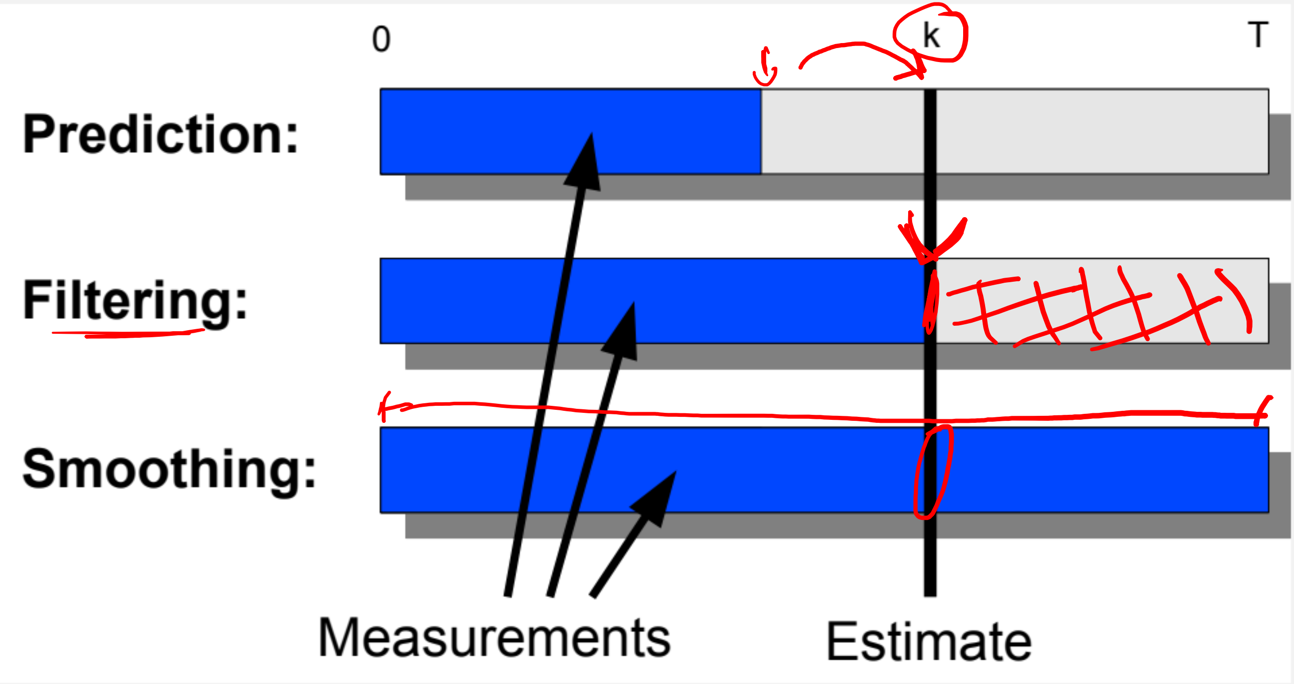
wrong



~~moiset~~
touin

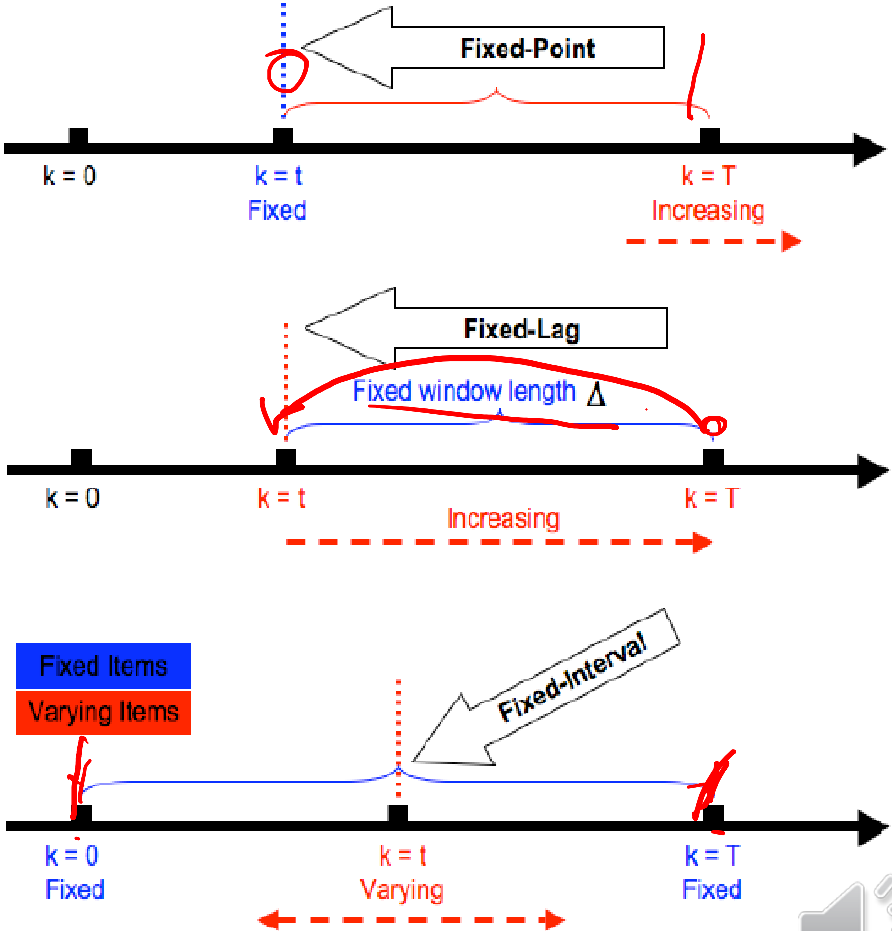
From: Chiang, K., Lin, Y., Huang, Y. et al. An ANN–RTS smoother scheme for accurate INS/GPS integrated attitude determination. GPS Solut 13, 199–208 (2009). https://doi.org/10.1007/s10291-008-0113-0

FILTERING/SMOOTHING



UNIVERSITÄT BONN igg

SMOOTHING



UNIVERSITÄT BONN igg

KALMAN FILTER

Predicted covariance

Predicted estimate

System noise

$$\mathbf{x}_k^- = \Phi \mathbf{x}_{k-1}$$

$$P_k^- = \Phi P_{k-1} \Phi^T + \mathbf{Q}$$

Prediction

measurement noise

Kalman Gain

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

$$\mathbf{x}_k = \mathbf{x}_k^- + K_k (z_k - H \mathbf{x}_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

Correction

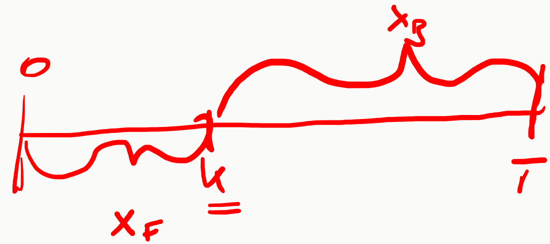
measurement

Predicted measurement

Updated estimate

Updated covariance

FORWARD BACKWARD FILTER



$$\hat{x}_k = K_{F_k}\hat{x}_{F_k} + K_{B_k}\hat{x}^-_{B_k}$$

$$= K_F x_F + (1-K_F)x_B^-$$

$$P(x) = E\left[(x-\hat{x})(x-\hat{x})^T\right]$$

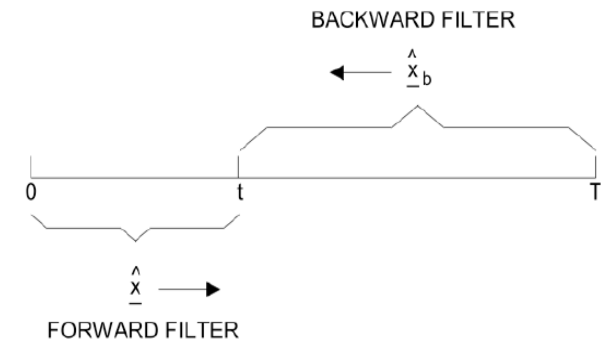
minimize P

$$\Rightarrow \quad K_F = \frac{P_B^-}{P_F + P_B^-} \qquad K_B = \frac{P_F}{P_F + P_B^-}$$

$$P = \left(P_F^{-1} + P_B^{-1}\right)^{-1}$$

weighted mean

FORWARD BACKWARD FILTER



$$
\begin{aligned}
\mathbf{x}_{S,k} &= K_{F,k}\mathbf{x}_{F,k} + K_{B,k}\mathbf{x}^{-}_{B,k} \\
&= K_{F,k}\mathbf{x}_{F,k} + (1 - K_{F,k})\mathbf{x}^{-}_{B,k} \\
K_{F,k} &= P^{-}_{R,k}(P_{F,k} + P^{-}_{B,k})^{-1} \\
P_{S,k} &= \left[(P_{F,k})^{-1} + (P^{-}_{B,k})^{-1}\right]^{-1}
\end{aligned}
$$

Details on the derivation in:
Dan Simon: Optimal State Estimation, Wiley Interscience

RAUCH-TUNG-STRIEBEL (RTS) SMOOTHER

1) $\hat{x}_F$, $\hat{P}_F$ $\quad 0 \longrightarrow T$

2) $T \longrightarrow 0$

$$\hat{x}_k = \hat{x}_{F_k} + A\left(\hat{x}_{k+1} - \hat{x}_{F_{k+1}}\right)$$

$$A = \frac{\hat{P}_{F_k} \cdot \Phi_k^T}{P^-_{F_{k+1}}}$$

UNIVERSITÄT BONN igg

RAUCH-TUNG-STRIEBEL (RTS) SMOOTHER

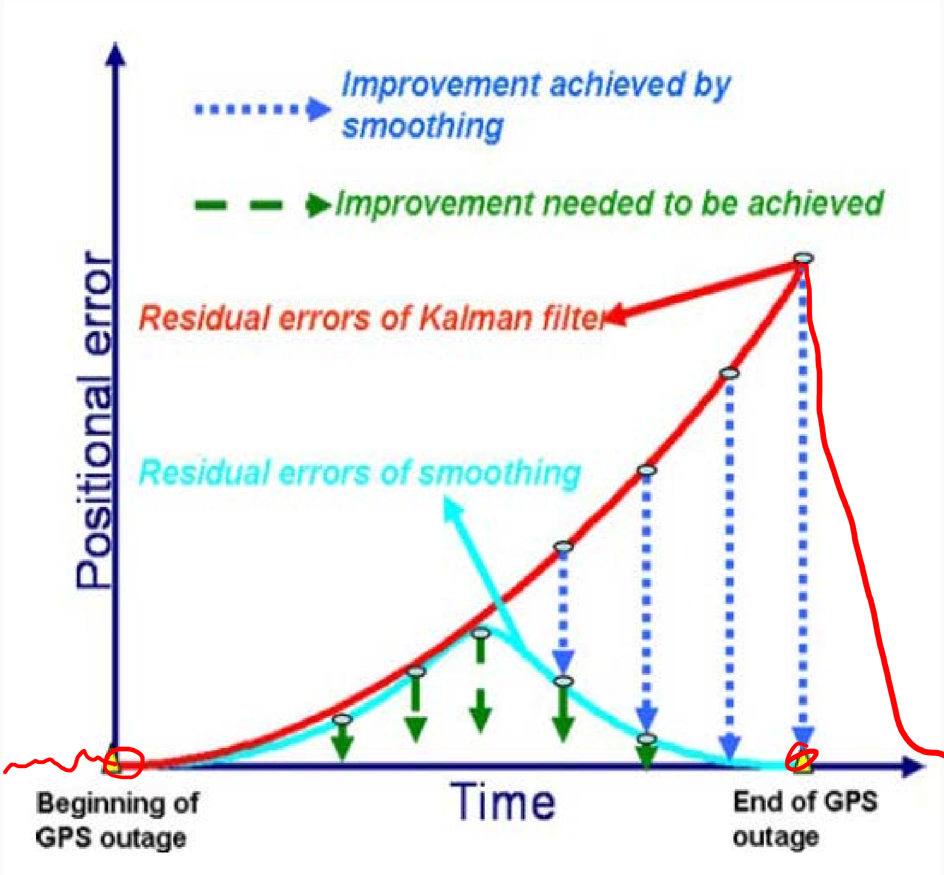
Runs from k to 1

$$\mathbf{x}_{S,k} = \mathbf{x}_{F,k} + A_k(\mathbf{x}_{S,k+1} - \mathbf{x}^-_{F,k+1})$$

$$A_k = P_{F,k}\Phi_k^T \left[P^-_{F,k+1}\right]^{-1}$$

$$P_{S,k} = P_{F,k} + A_k(P_{S,k+1} - P^-_{F,k+1})$$

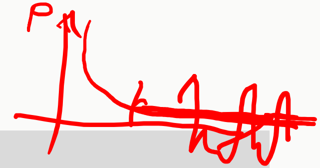
Details on the derivation in:
Dan Simon: Optimal State Estimation, Wiley Interscience



From: Chiang, K., Lin, Y., Huang, Y. et al. An ANN–RTS smoother scheme for accurate INS/GPS integrated attitude determination. GPS Solut 13, 199–208 (2009). https://doi.org/10.1007/s10291-008-0113-0

HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

A) LOOK AT THE COVARIANCE OF THE ESTIMATE



Prediction

$$\mathbf{x}_k^- = \mathbf{\Phi}\mathbf{x}_{k-1}$$

$$P_k^- = \mathbf{\Phi} P_{k-1} \mathbf{\Phi}^T + \mathbf{Q}$$

Correction

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$

$$\mathbf{x}_k = \mathbf{x}_k^- + K_k (z_k - H\mathbf{x}_k^-)$$

$$P_k = (I - K_k H) P_k^-$$

- Does not depend on measurements
- Only ‚valid‘, if all models and uncertaincies are ‚correct‘
- No systematic errors included

HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

B) LOOK AT THE INNOVATION

Prediction

$$\mathbf{x}_k^- = \mathbf{\Phi}\mathbf{x}_{k-1}$$
$$P_k^- = \mathbf{\Phi} P_{k-1} \mathbf{\Phi}^T + \mathbf{Q}$$

Correction

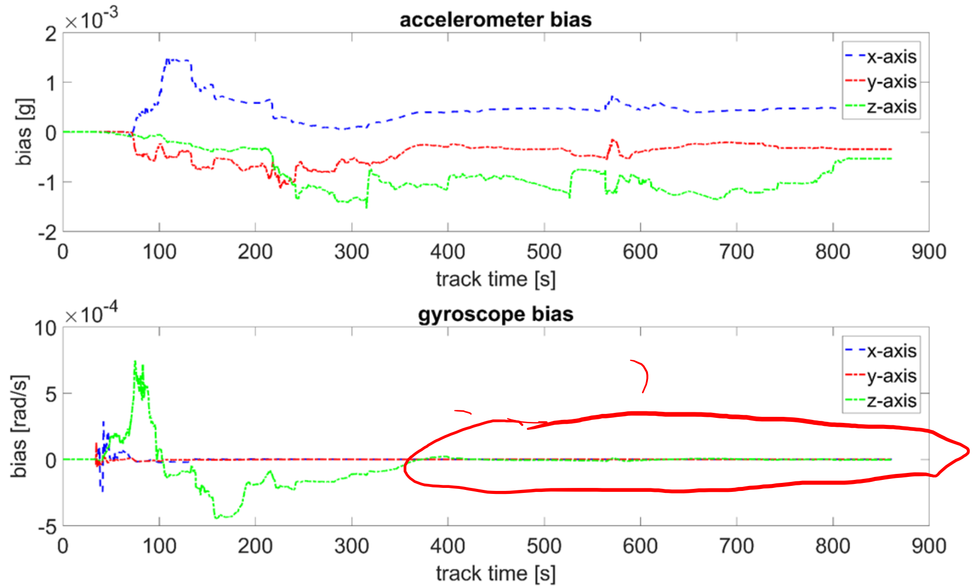
$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$
$$\mathbf{x}_k = \mathbf{x}_k^- + K_k(\underbrace{z_k - H\mathbf{x}_k^-}_{\text{Innovation}})$$
$$P_k = (I - K_k H)P_k^-$$

- Gives an idea about the consistency between models and observations
- It is not clear if iconsistencies come from wrong measurements or predictions

UNIVERSITÄT BONN igg

HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

C) CAREFULLY LOOK AT ALL ESTIMATED STATES

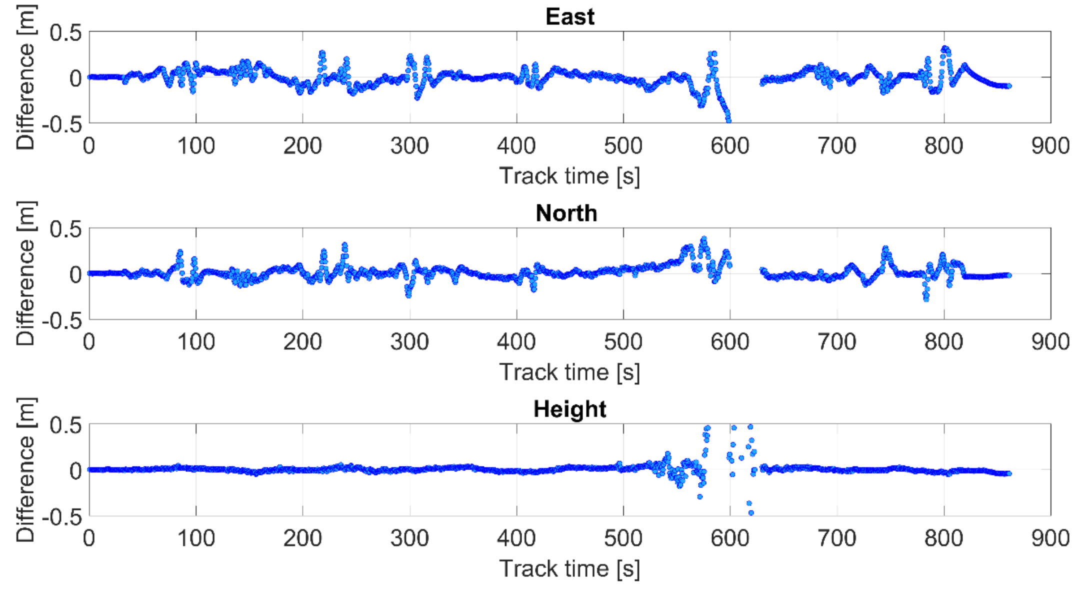


- A nice looking position result can also appear if your process model is completely wrong and your measurements are good
- E.g. Estimated bias values show ‚consistency‘ of models

UNIVERSITÄT BONN igg

HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

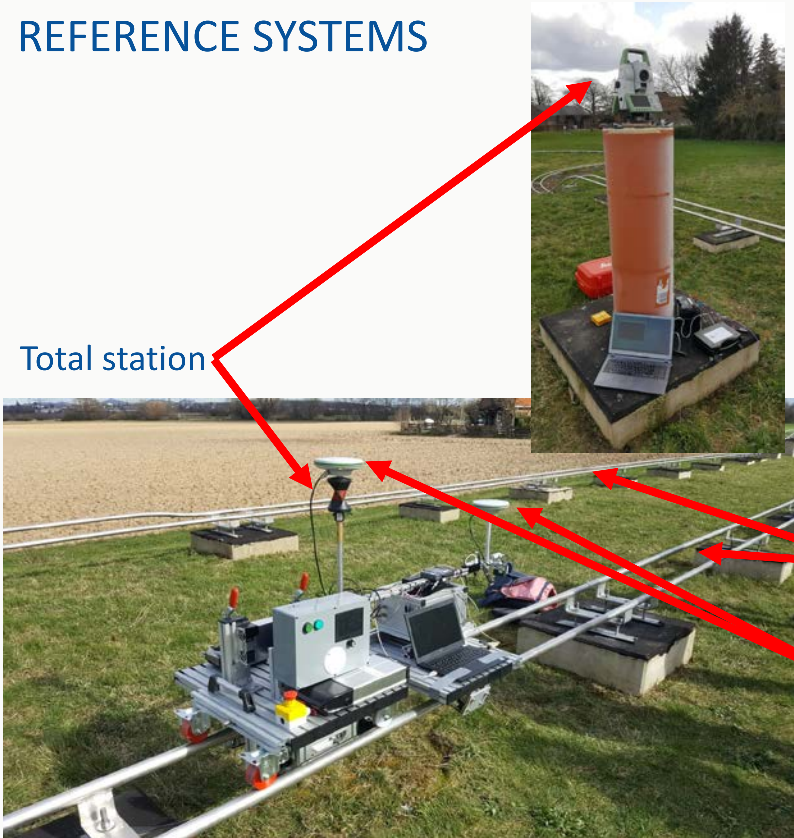
D) COMPARE WITH REFERENCE SOLUTION



- Needs synchronization between reference and estimation results
- Difficult to realize in outdoor scenarios (especially for orientation)
- How to ‚parametrize' the error? Depends on motion, environment, time …

HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

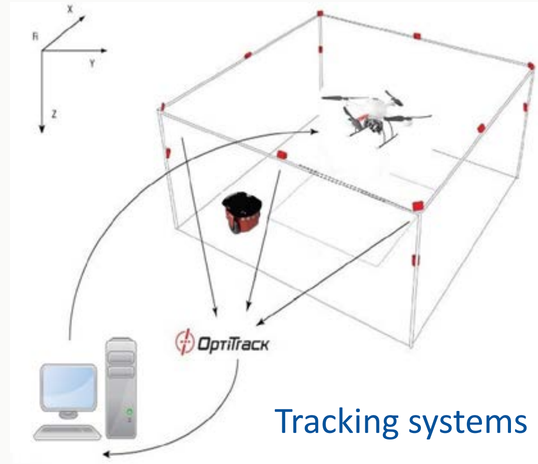
REFERENCE SYSTEMS



Total station

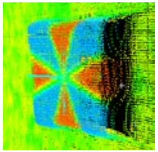
Reference trajectory

RTK GNSS



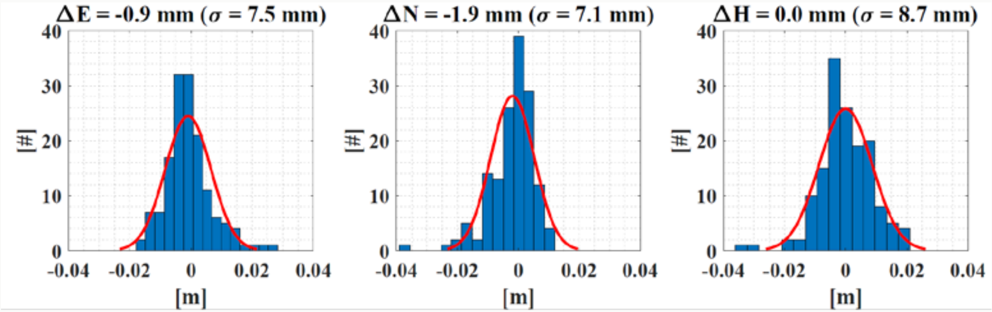
HOW TO EVALUATE THE QUALITY OF THE ESTIMATION?

E) INDIRECT EVALUATION



$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} + \mathbf{R}_n^e(L,B)\,\mathbf{R}_b^n(\phi,\theta,\psi)\cdot\left[\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} + \mathbf{R}_s^b(\alpha,\beta,\gamma)\cdot\begin{bmatrix} 0 \\ d\cdot\sin b \\ d\cdot\cos b \end{bmatrix}\right]$$

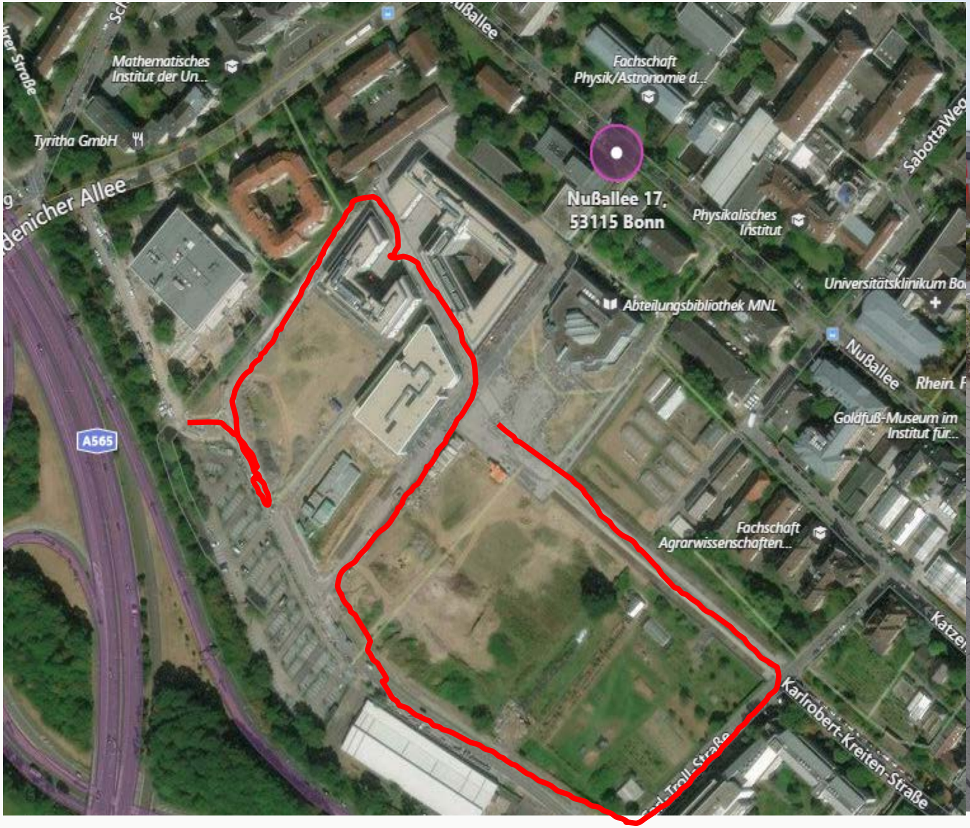
Compare with known reference



- Includes many other errors
- Assignement of error to specific sources is difficult

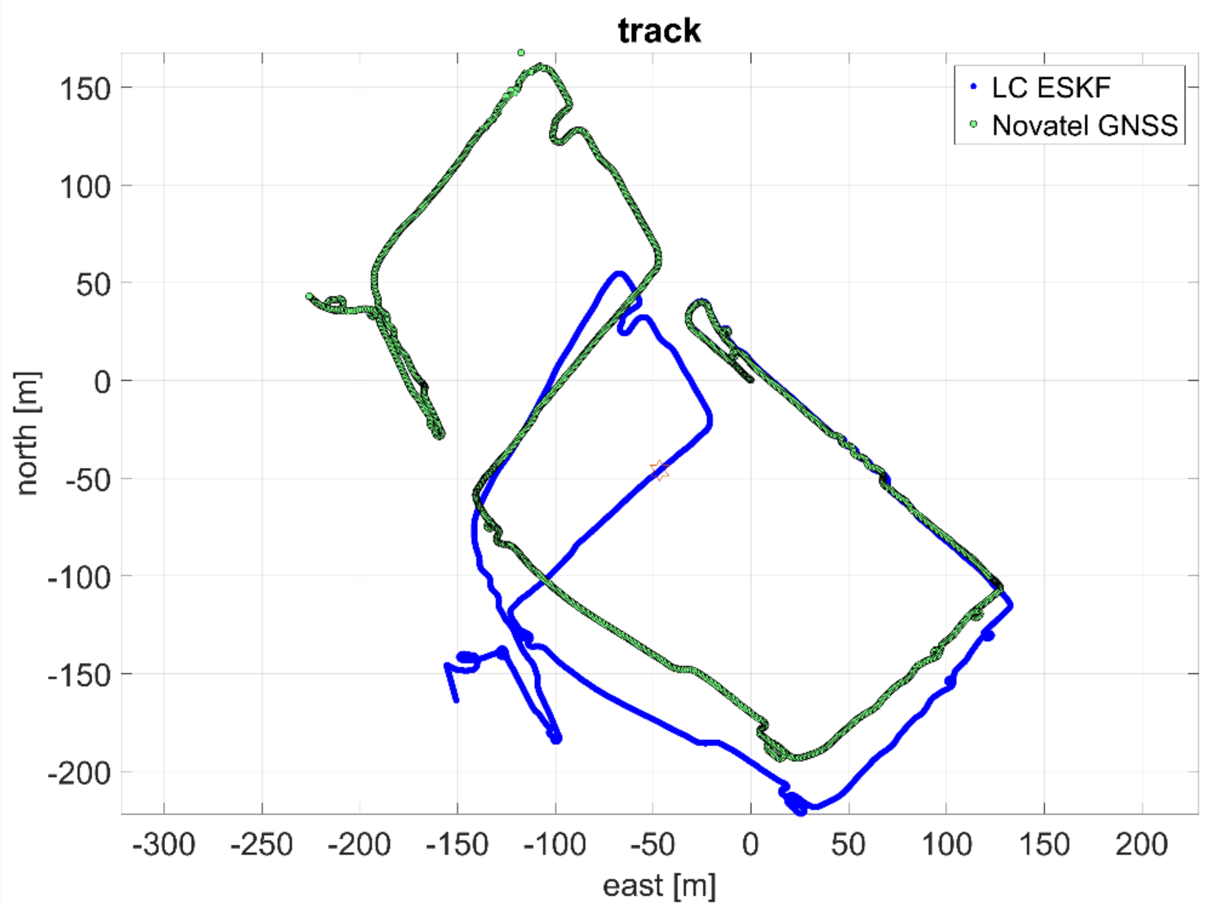
UNIVERSITÄT BONN igg

EXAMPLE



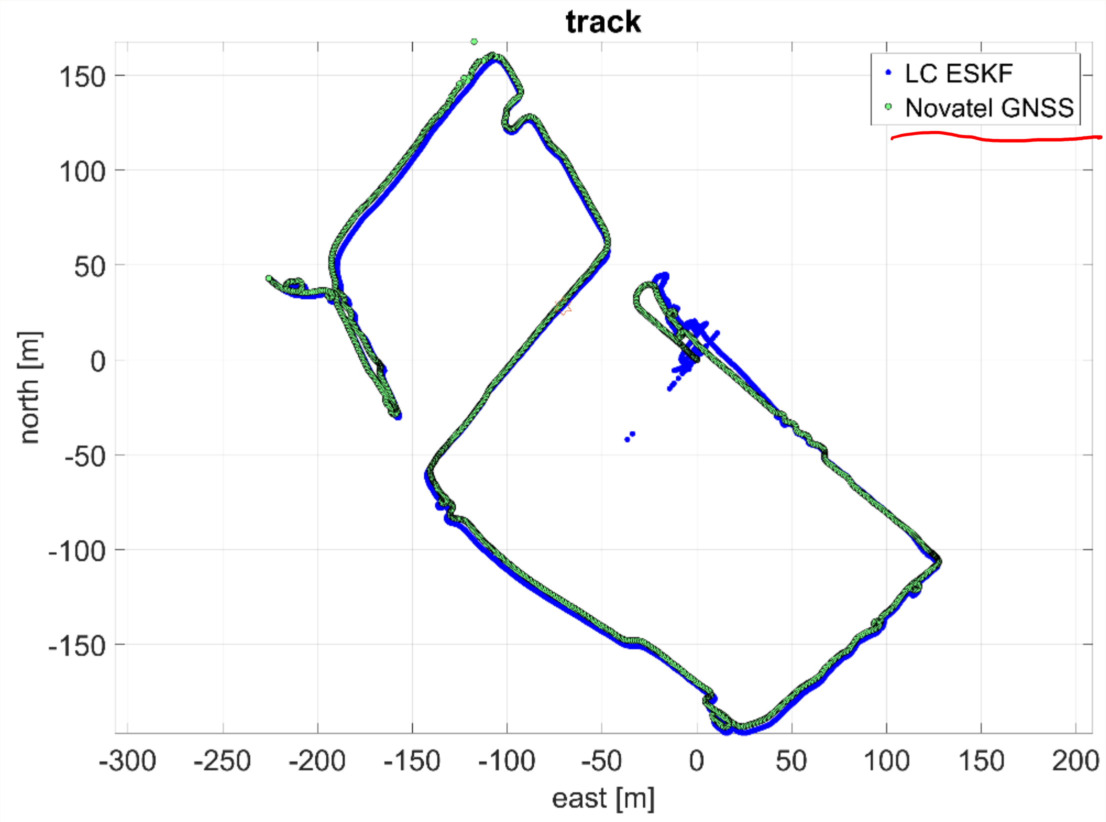
UNIVERSITÄT BONN igg

STRAPDOWN ONLY

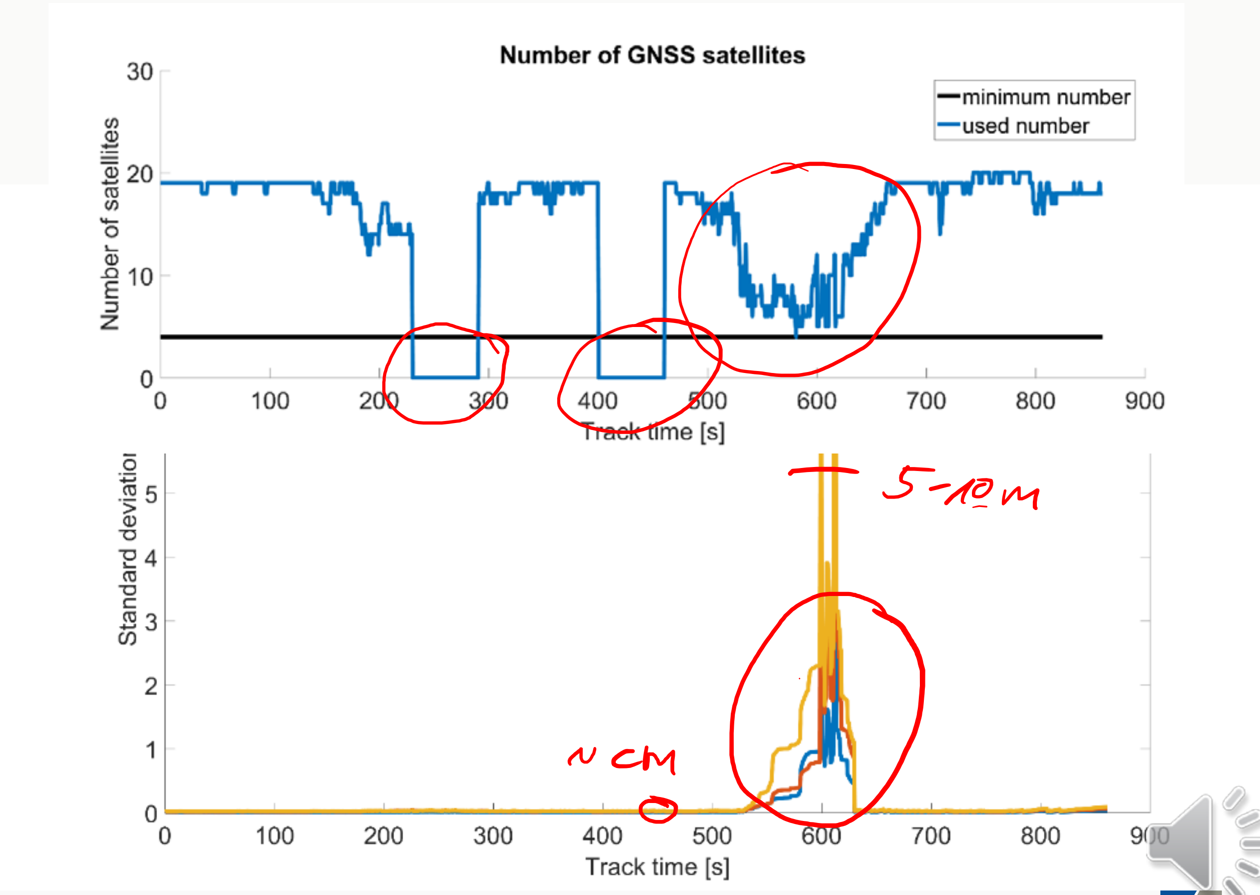


UNIVERSITÄT BONN
igg

KF WITH ODOMETER. NO GPS!

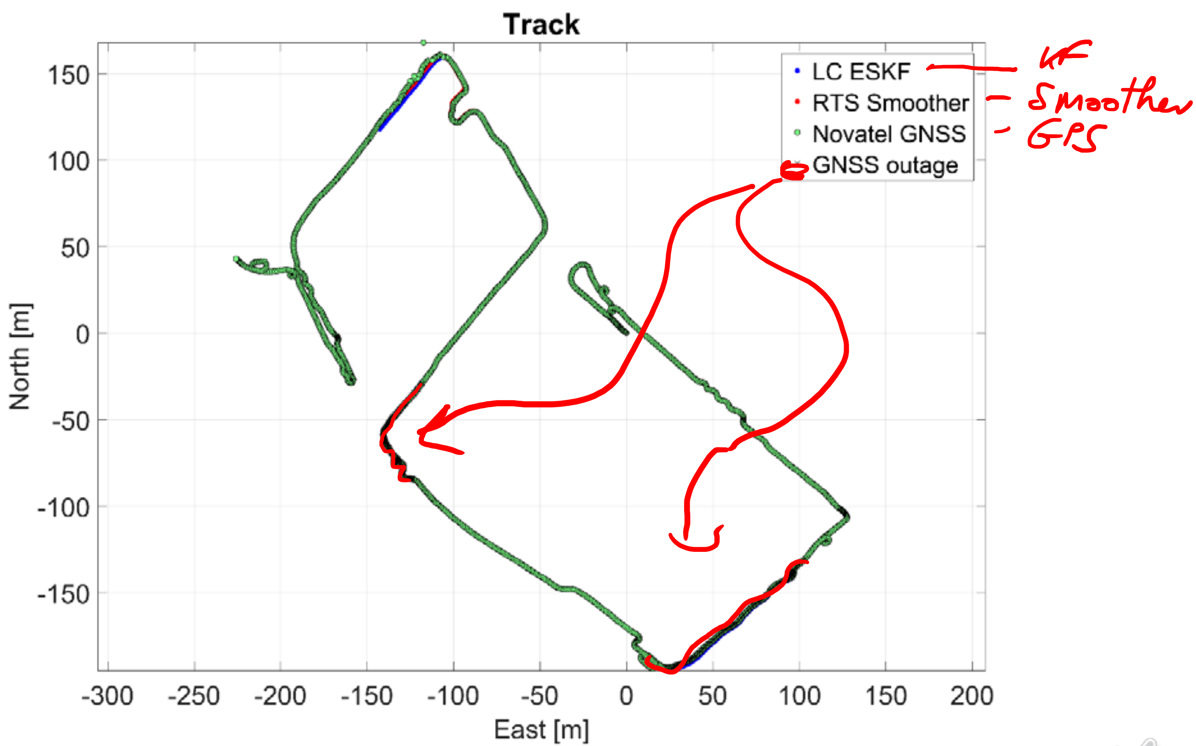


GNSS QUALITY



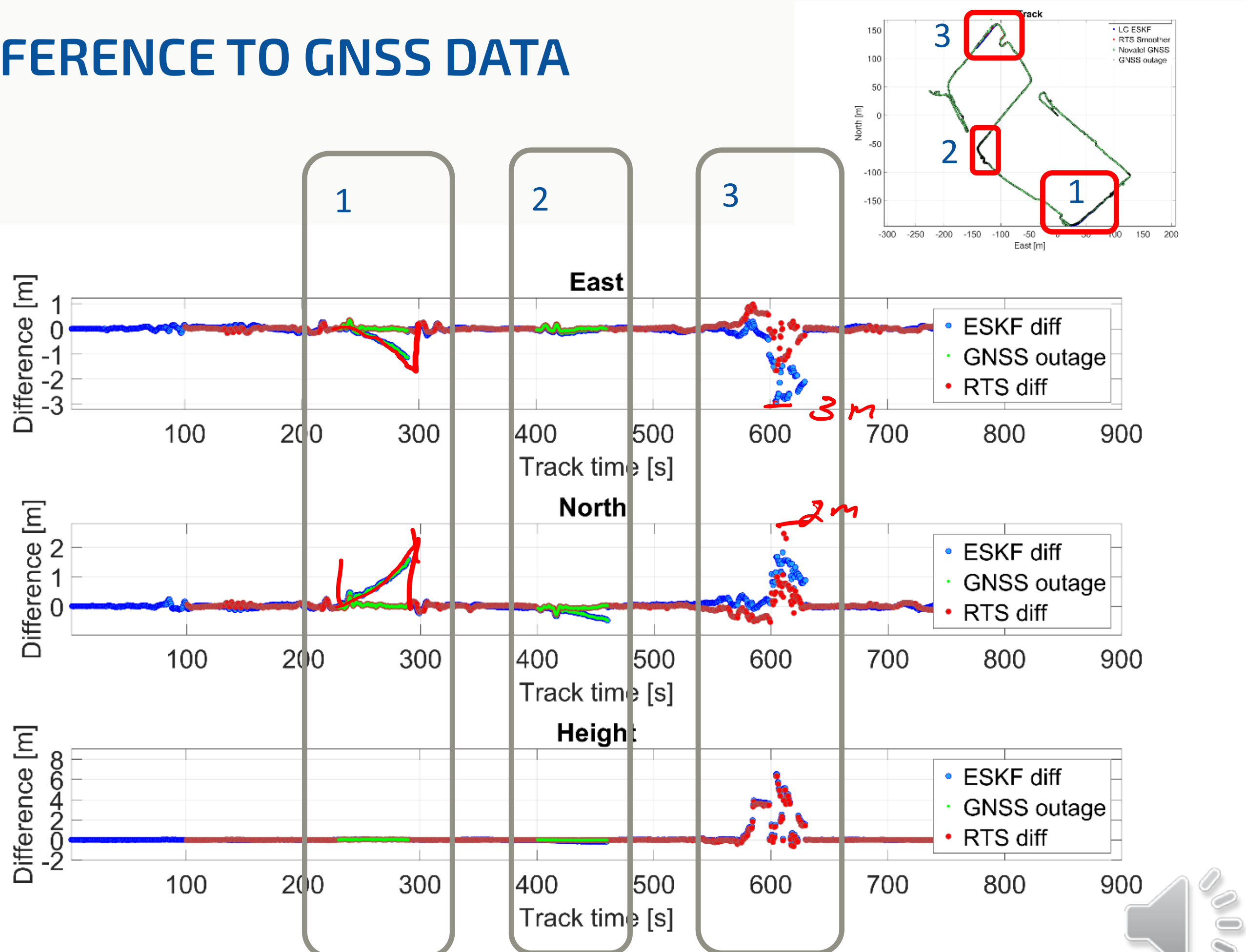
UNIVERSITÄT BONN
igg

TRACK

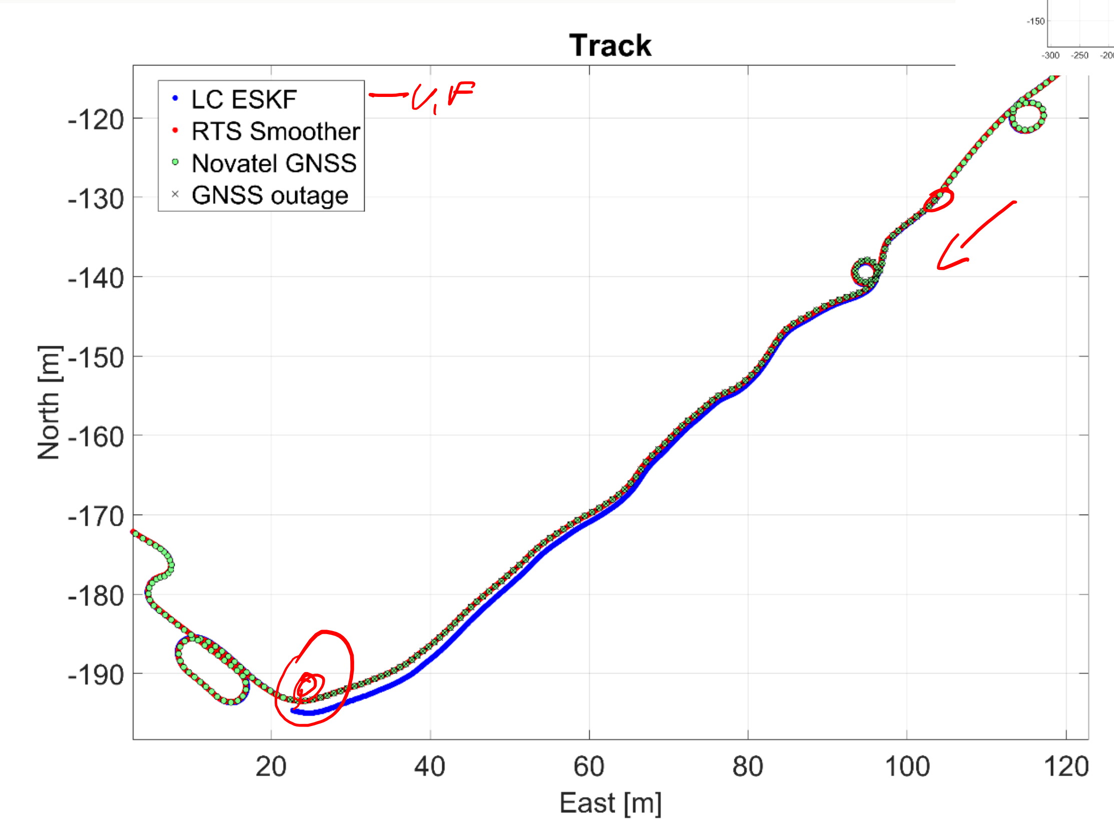


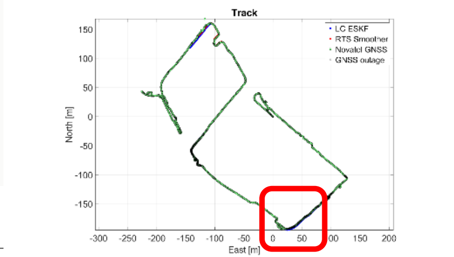
Track comparison: RTS smoother without and with GNSS outages

DIFFERENCE TO GNSS DATA



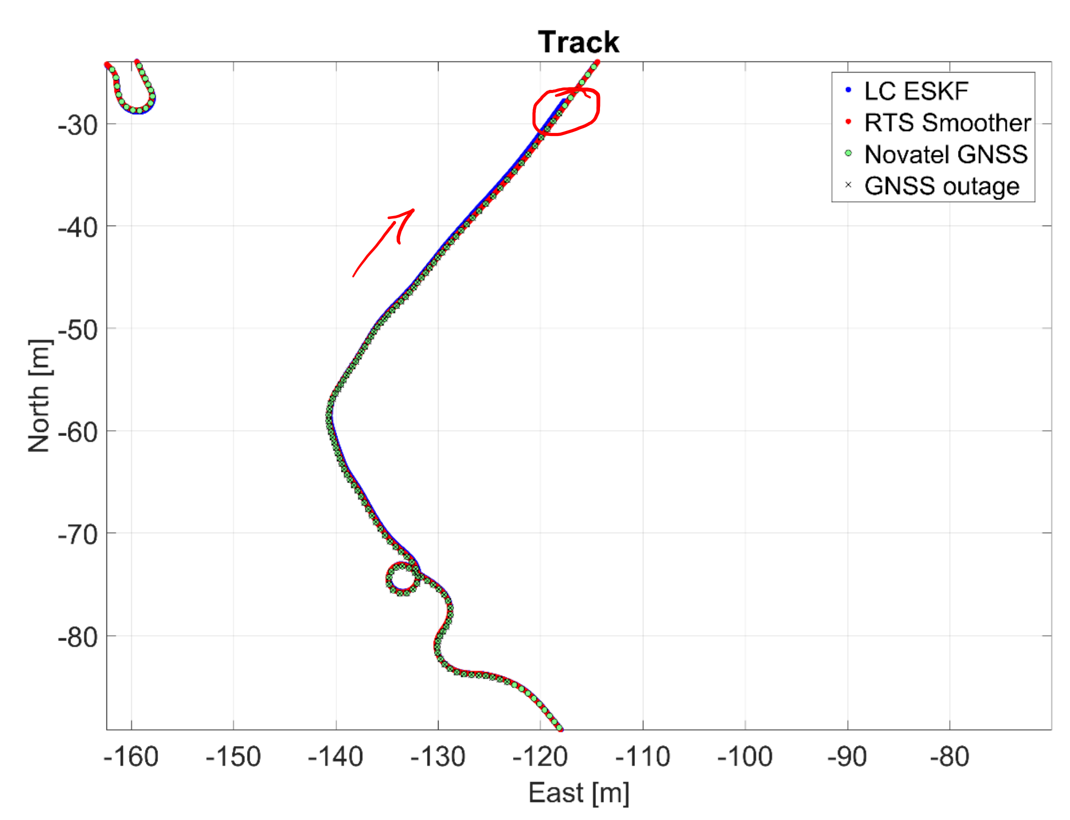
ZOOM 1



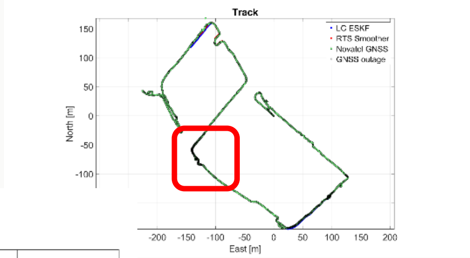


UNIVERSITÄT BONN igg

ZOOM 2

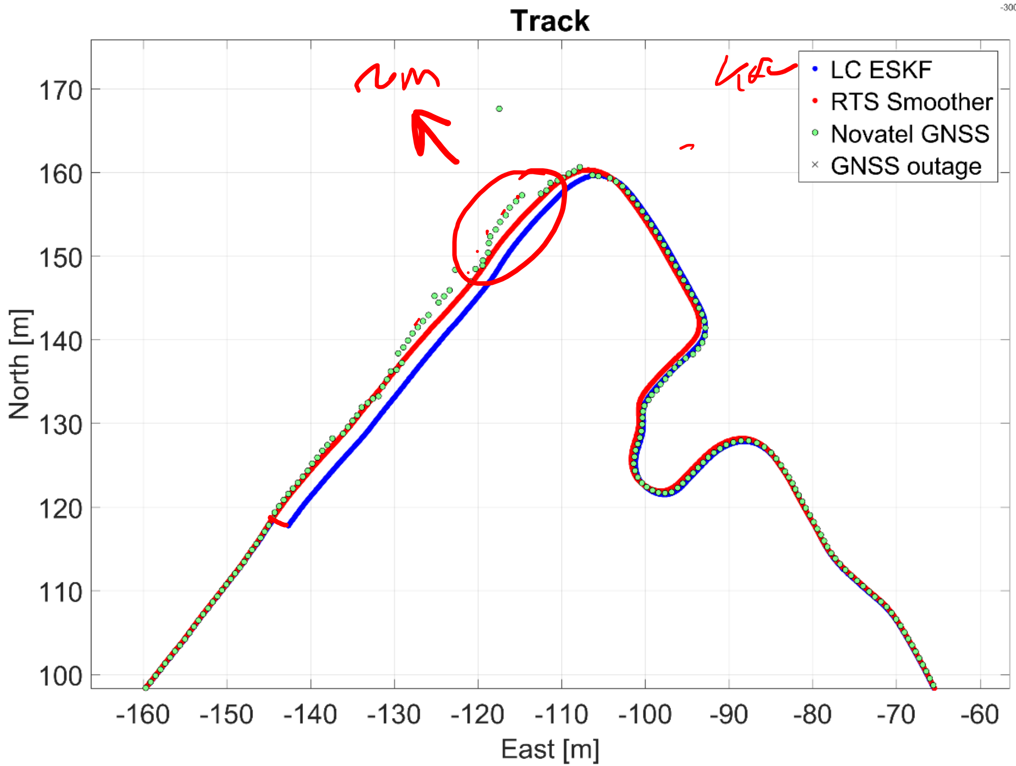


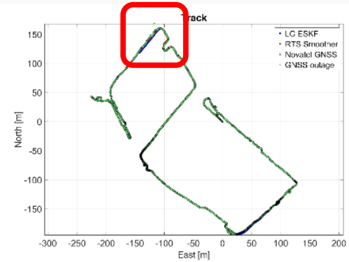
Track
LC ESKF
RTS Smoother
Novatel GNSS
GNSS outage
North [m]
East [m]
-30
-40
-50
-60
-70
-80
-160
-150
-140
-130
-120
-110
-100
-90
-80



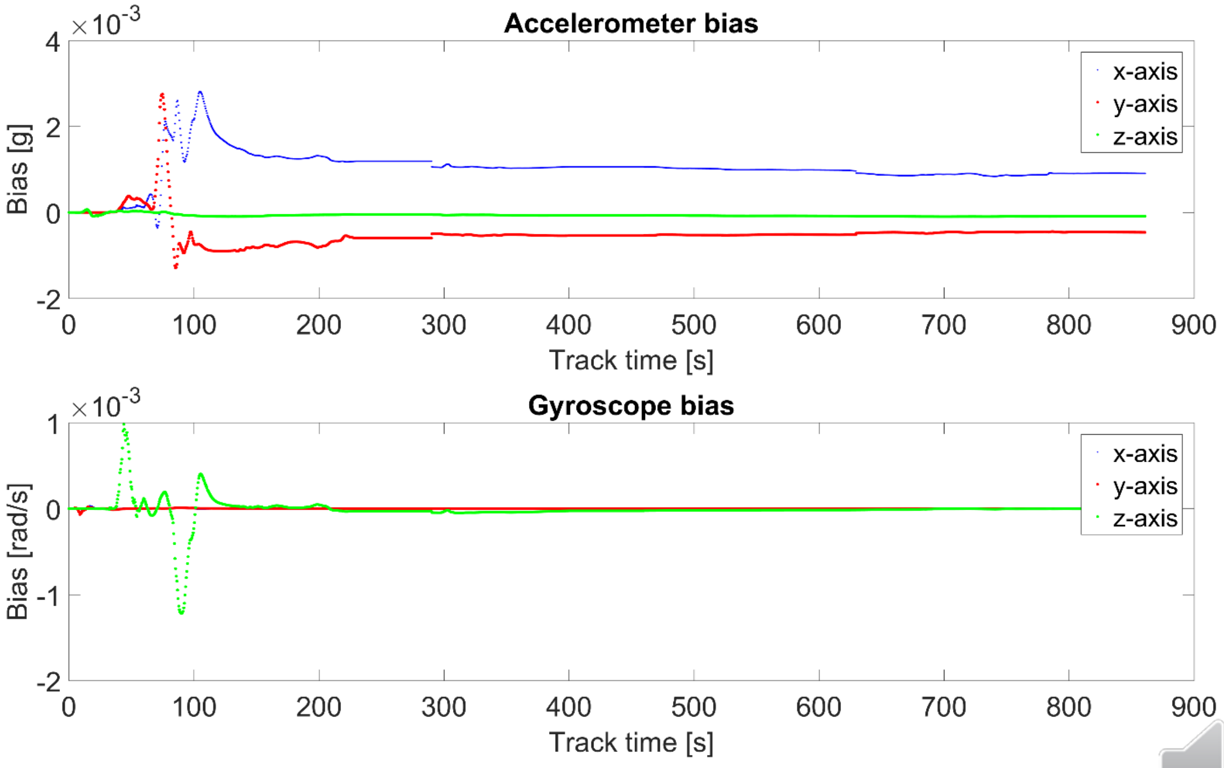
Track
LC ESKF
RTS Smoother
Novatel GNSS
GNSS outage
North [m]
East [m]

ZOOM 3


Track
LC ESKF
RTS Smoother
Novatel GNSS
GNSS outage
North [m]
East [m]


Track
LC ESKF
RTS Smoother
Novatel GNSS
GNSS outage
North [m]
East [m]

BIAS ESTIMATION



WHAT YOU HAVE LEARNED TODAY

- How are position and ortientation calculated using inertial sensors and GNSS?
- How are inertial sensors and GNSS observations combined using filtering algorithms?
- How are inertial sensor errors influencing the estimation?
- What is the difference between filtering and smoothing and when/why is smoothing better?
- What is the RTS smoother?
- How can you evaluate the quality of a trajectory estimation algorithm/system?

THANKS